Guide to the Systems Engineering Body of Knowledge (SEBoK) v1.4
-
Part 7: SE Implementation Examples

BKCASE

June 29, 2015

Contents

Cover

Letter from the Editor

A very warm welcome to all SEBoK users, both old and new. The BKCASE Editor in Chief (EIC) has overall responsibility for the continuing review and update of the SEBoK. Many thanks to the BKCASE Governors and the current members of the Editorial Board for supporting me.

I am delighted to be able to talk to you about SEBoK v. 1.4 which continues our commitment to regular review of the information referenced in our "Guide to the Systems Engineering Body of Knowledge".

This new version features changes to respond to publication of ISO/IEC/IEEE. 15288:2015 Systems and Software Engineering - System Life Cycle Processes and the INCOSE SE Handbook v4.0, 2015. Over the last 12 months the BKCASE Editorial Board has made significant efforts to become more involved in activities within our sponsoring organizations on key topics such as model based systems engineering (MBSE), systems of systems, systems engineering leadership, etc. You will begin to see the impact of this in v1.4 of the SEBoK, with further updates in v. 1.5, planned for Autumn 2015.

SEBoK v. 1.4

SEBoK v.1.4 feels like something of a turning point for the body of knowledge. On one hand we have “closed the loop” on the current cycle of updates to our core Systems Engineering reference sources, while on the other we have taken the first steps towards a richer relationship with other sources of knowledge and turned our focus onto some of the exciting transformations happening within Systems Engineering.

A brief summary of the changes in this version are given below, for details of content affected by these updates go to Acknowledgements and Release History.

A small but significant change has been made in SEBoK Part 2. This has been renamed from Systems to Foundations of Systems Engineering. This change reflects the focus of part 2 on the wider knowledge sources which underpin or enable good SE practice. While this has always been our aim for part 2, the old name was interpreted as an overview of all systems knowledge by some both inside and outside of BKCASE. This confusion led to a confusion in scope and purpose of some articles and miss understanding of our relationships with the systems science community. With this change we have begun to address this miss understanding and provide a firmer basis for this part of the SEBoK in future.

The most significant change to the SEBoK for v1.4 is in Part 3: SE and Management. A number of the technical and project process articles in SEBoK have been updated to reflect the revisions of ISO/IEC/IEEE 15288 (ISO 2015):

- A new “**Business or Mission Analysis Process**” has been added to the standard. This process defines SE activities to assist business or enterprise decision makers to define the problem space, identify the stakeholders, develop preliminary operational concepts, and distinguish environmental conditions and constraints that bound the solution space. This process follows the same approach as the Business or Mission Analysis article which was already in SEBoK. This article has been updated to better align with the standard.
- The “**Stakeholder Requirement Definition Process**” in the standard has been renamed “**Stakeholder Needs and Requirements Definition**”. The revised process builds on the change above to include more description of how to translate stakeholder needs and business strategy into requirements. The SEBoK article Stakeholder Needs and Requirements has been updated to better align with the standard; a new article Life Cycle Processes and Enterprise Need has been added to discuss how requirements can be related to business strategy and needs where appropriate.
- The “**Architectural Design Process**” in the standard has been replaced with an “**Architecture Definition Process**” which focuses more on the identification of stakeholder concerns and the higher level system architecture that will address the concerns. A new “**Design Definition Process**” describes how system architecture translates into realisable system design. Two new SEBoK articles System Architecture and System Design reflect

this revision of the standard and replace the previous article on architectural design. The Systems Requirements article has also changed to reflect these updates.

- The Logical Architecture Model Development and Physical Architecture Model Development SEBoK articles remain, describing the development of a couple commonly used architecture models in more detail as described in other standards, such as ISO/IEC/IEEE 42010, Systems and Software Architecture Description (ISO 2011).
- A new "**System Analysis Process**" has been added to the standard. This process defines SE activities to allow developers to objectively plan and carry out quantitative assessments of a system or aspects of a system, in order to select and/or update the most efficient system architecture and to generate derived engineering data. This process follows the same approach as the System Analysis article which was already in SEBoK. This article has been updated to better align with the standard.

Some of the changes to the standard build on the descriptions of SE which were developed for the SEBoK. This is not surprising, since many of the same authors were involved in both. The alignment of these views also includes the updated INCOSE SE Handbook v4.0 (INCOSE 2015), which now fully aligns with the standard. This completes a process which has overlapped the creation of the SEBoK. Going forward we plan to expand the scope of knowledge in the SEBoK to cover broader applications of SE within this generic framework of Life Cycle Processes. It is likely that this will shift the focus of activity from SEBoK Part 3 to SEBoK Part 4: Applications of SE and in particular the tailoring of SE to a range of application types and domains.

Some other small changes have been made in Parts 4, 5 and 7 as part of the ongoing review of SEBoK material to reflect new source material.

Future Direction for SEBoK

Once again, many thanks to the "core group of dedicated and knowledgeable contributing authors and reviewers" who make up the BKCASE community. It has been my privilege over the last 12 months to continue working with and grow this community and to expand our relationships with key organizations and groups both within systems engineering and outside of it.

The role of the Editorial Board is to work with this community of interest on an ongoing review of the current SEBoK content and structure and to develop plans for its maintenance and evolution. Our overall goals in evolving the SEBoK remain broadly the same as those outlined in the previous SEBoK updates. I have restated and slightly modified those goals below:

- Improve the ways in which Part 1 (SEBoK Introduction) provides a starting point for different SEBoK users to find and navigate knowledge relevant to them. This will include consideration of some of the SEBoK Use Cases which were not expanded in previous releases, and possible new case studies covering application domains such as Defense, Health Care or Transport.
- Review Part 2 (Foundations of Systems Engineering) with help from the International Society for the Systems Sciences (ISSS) to better understand the relationships between Systems Science (glossary) and Systems Thinking (glossary) as applied to engineered systems. We hope this will lead to an improved integration of systems principles, concepts, patterns and models into the other systems engineering focused knowledge areas across the SEBoK.
- Look for broader views on the key practices of Part 3 (Systems Engineering and Management) to feed back into the ongoing co evolution of key standards. In particular make more direct reference to the continuing evolution of Agile life cycle thinking and bring in more knowledge sources from the model based SE (MBSE) community.
- Expand our coverage of knowledge on systems engineering application and practices. In particular look for ways to bring in more knowledge on how systems engineering practices such as architecting, life cycle tailoring and model based systems engineering are applied in other domains.
- Identify the other groups, both within the systems engineering community and beyond, with interest in the topics of Part 5 (Enabling Systems Engineering) and Part 6 Related Disciplines and form stronger relationships with

them.

We continue to work towards ensuring that our coverage of existing systems engineering knowledge is complete and to push the boundaries of that knowledge into new approaches and domains. I also want to strengthen further our links to all members of the systems engineering community through things like the SEBoK Sandbox. If you are interested in any of the activity discussed above or if you have other topics which we should be considering please contact me or the appropriate member of the Editorial Board directly or use one of the available feedback mechanisms.

We have made a good start on gathering review comments and content suggestions from as wide a variety of individuals as possible to make the SEBoK a truly community-led product. Thank you to all those who have already joined this effort and I continue to look forward to working with many of you on future SEBoK releases.

Thank you,

BKCASE Governance and Editorial Board

BKCASE Governing Board

The three SEBoK steward organizations – the International Council on Systems Engineering (INCOSE), the Institute of Electrical and Electronics Engineers Computer Society (IEEE-CS), and the Systems Engineering Research Center (SERC) provide the funding and resources needed to sustain and evolve the SEBoK and make it available as a free and open resource to all. The stewards appoint the BKCASE Governing Board to be their primary agents to oversee and guide the SEBoK and its companion BKCASE product, GRCSE.

The BKCASE Governing Board includes:

- **INCOSE**
  - Paul Frenz, William Miller (Governing Board Chair)
- **IEEE Computer Society**
  - Richard Fairley, Massood Towhidnejad
- **SERC**
  - Art Pyster, Cihan Dagli

Past INCOSE governors Kevin Forsberg, David Newbern, David Walden, Courtney Wright, Dave Olwell, and Ken Nidiffer. The governors would also like to acknowledge John Keppler, IEEE Computer Society, who has been instrumental in helping the Governors to work within the IEEE CS structure.

The stewards appoint the BKCASE Editor in Chief to manage the SEBoK and GRCSE and oversee the Editorial Board.

Editorial Board

The SEBoK Editorial Board is chaired by an Editor in Chief, supported by a group of Associate Editors.

BKCASE Editor in Chief

Richard D. Adcock

Cranfield University (UK)

richard.adcock@incose.org [1]

Responsible for the appointment of SEBoK Editors and for the overall content and coherence of the SEBoK.

Each Editor has his/her area(s) of responsibility, or shared responsibility, highlighted in the table below.

SEBoK Part 1 SEBoK Introduction

Ariela Sofer

George Mason University (USA)

asofer@gmu.edu [2]

Responsible for Part 1

SEBoK Part 2: Systems

Cihan Dagli

Missouri University of Science & Technology (USA)

dagli@mst.edu [3]

Responsible for the Systems Approach Applied to Engineered Systems knowledge areas

Dov Dori

Massachusetts Institute of Technology (USA) and Technion Israel Institute of Technology (Israel)

dori@mit.edu [4]

Responsible for the Representing Systems with Models knowledge area

Duane Hybertson

MITRE (USA)

Jointly responsible for the Systems Fundamentals, Systems Science and Systems Thinking knowledge areas

Janet Singer (USA)

Jointly responsible for the Systems Fundamentals, Systems Science and Systems Thinking knowledge areas

Mike Yearworth

University of Bristol (UK)

Jointly responsible for the Systems Fundamentals, Systems Science and Systems Thinking knowledge areas

SEBoK Part 3: Systems Engineering and Management

Barry Boehm

University of Southern California (USA)

boehm@usc.edu [5]

Jointly responsible for the Systems Engineering Management and Life Cycle Models knowledge areas

Kevin Forsberg

OGR Systems

Jointly responsible for the Systems Engineering Management and Life Cycle Models knowledge areas

Gregory Parnell

University of Arkansas (USA)

gparnell@uark.edu [6]

Responsible for Systems Engineering Management knowledge area.

Garry Roedler

Lockheed Martin (USA)

garry.j.roedler@lmco.com [7]

Responsible for the Concept Definition and System Definition knowledge areas.

Ricardo Valerdi

University of Arizona (USA)

rvalerdi@email.arizona.edu [8]

Responsible for the System Realization knowledge area.

Ken Zemrowski

TASC

kenneth.zemrowski@incose.org [9]

Responsible for the Systems Engineering Standards knowledge area.

SEBoK Part 4: Applications of Systems Engineering

Judith Dahmann

MITRE Corporation (USA)

jdahmann@mitre.org [10]

Jointly responsible for Product Systems Engineering and Systems of Systems (SoS) knowledge areas

Rick Hefner

California Institute of Technology (USA)

Responsible for the Service Systems Engineering knowledge area.

Michael Henshaw

Loughborough University (UK)

M.J.d.Henshaw@lboro.ac.uk [11]

Jointly responsible for Product Systems Engineering and Systems of Systems (SoS) knowledge areas

James Martin *The Aerospace Corporation*

james.martin@incose.org [12]

Responsible for the Enterprise Systems Engineering knowledge area.

SEBoK Part 5: Enabling Systems Engineering

Heidi Davidz

Aerojet Rocketdyne (USA)

heidi.davidz@rocket.com [13]

Jointly responsible for the Enabling Individuals and Enabling Teams knowledge area

Emma Sparks

Cranfield University

Jointly responsible for the Enabling Individuals and Enabling Teams knowledge area

SEBoK Part 6 Related Disciplines

Alice Squires

Washington State University (USA)

alice.squires@wsu.edu [14]

Responsible for Part 6

SEBoK Part 7 Systems Engineering Implementation Examples

Brian Sauser

University of North Texas (USA)

brian.sauser@unt.edu [15]

Responsible for Part 7: Systems Engineering Implementation Examples, which includes Case Studies and Vignettes

Brian White

CAU>SE (USA)

bewhite71@gmail.com [16]

Responsible for Part 7: Systems Engineering Implementation Examples, which includes Case Studies and Vignettes

Graduate Reference Curriculum for Systems Engineering (GRCSE)

David H. Olwell

Naval Postgraduate School (USA)

Senior Editor for GRCSE.

Editorial Board Support

The Assistant Editor provide general editorial support across all topics and assist with both content improvement and production issues.

BKCASE Assistant Editor

Claus Ballegaard Nielsen

Cranfield University (UK)

c.nielsen@cranfield.ac.uk [17]

Interested in Editing?

The Editor in Chief is looking for additional editors to support the evolution of the SEBoK. Editors are responsible for maintaining and updating one to two knowledge areas, including recruiting and working with authors, ensuring the incorporation of community feedback, and maintaining the quality of SEBoK content. We are specifically interested in support for the following knowledge areas:

- System Deployment and Use
- Product and Service Life Management
- Enabling Businesses and Enterprises
- Systems Engineering and Software Engineering
- Systems Engineering and Procurement/Acquisition
- Systems Engineering and Specialty Engineering

If you are interested in being considered for participation on the Editorial Board, please visit the BKCASE website http://www.bkcase.org/join-us/or contact the BKCASE Staff directly at bkcase.incose.ieeecs@gmail.com [18].

SEBoK v. 1.4 released 29 June 2015

SEBoK Discussion

Please provide your comments and feedback on the SEBoK below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account. Simply type your comment in the text field below and DISQUS will guide you through the login or registration steps. Feedback will be archived and used for future updates to the SEBoK. *If you provided a comment that is no longer listed, that comment has been adjudicated. You can view adjudication for comments submitted prior to SEBoK v. 1.0 at SEBoK Review and Adjudication. Later comments are addressed and changes are summarized in the Letter from the Editor and Acknowledgements and Release History.*

If you would like to provide edits on this article, recommend new content, or make comments on the SEBoK as a whole, please see the SEBoK Sandbox [19].

ENCODED_CONTENT
MTkwMDgPGRpdiBpZD0iZGlzcXVzX3RocmVhZCI+PC9kaXY+CjxzY3JpcHQgdHlwZT0idGV4dC9qYXZhc2Nya
END_ENCODED_CONTENT

References

[1] mailto:richard.adcock@incose.org
[2] mailto:asofer@gmu.edu
[3] mailto:dagli@mst.edu
[4] mailto:dori@mit.edu
[5] mailto:boehm@usc.edu
[6] mailto:gparnell@uark.edu
[7] mailto:garry.j.roedler@lmco.com
[8] mailto:rvalerdi@email.arizona.edu
[9] mailto:kenneth.zemrowski@incose.org
[10] mailto:jdahmann@mitre.org
[11] mailto:M.J.d.Henshaw@lboro.ac.uk
[12] mailto:james.martin@incose.org
[13] mailto:heidi.davidz@rocket.com
[14] mailto:alice.squires@wsu.edu
[15] mailto:brian.sauser@unt.edu
[16] mailto:bewhite71@gmail.com
[17] mailto:c.nielsen@cranfield.ac.uk
[18] mailto:bkcase.incose.ieeecs@gmail.com
[19] http://www.sebokwiki.org/sandbox/

Acknowledgements and Release History

This article describes the contributors to the current version of the SEBoK. For information on contributors to past versions of the SEBoK, please follow the links under "SEBoK Release History" below. To learn more about the updates to the SEBoK for v. 1.4, please see the Letter from the Editor.

Governance

The SEBoK is shaped by the BKCASE Editorial Board and is overseen by the BKCASE Governing Board. A complete list of members for each of these bodies can be found on the BKCASE Governance and Editorial Board page.

Content and Feature Updates for 1.4

This is minor update, including changes related to ISO/IEC/IEEE 15288:2015 standard, updated articles in the areas of System Architecture, Life-Cycle processes, System of Systems, Competencies, Ethics and MBSE, as well as three new case studies.

For more information about this release please refer to Version 1.4.

SEBoK Release History

There have been 13 releases of the SEBoK to date, collected into 4 main releases.

- Version 1.0 – The first version intended for broad use.
- Version 1.1 - A minor update that made modest content improvements.
- Version 1.2 - A minor update, including two new articles and revision of several existing articles.
- Version 1.3 - A minor update, including three new case studies, a new use case, updates to several existing articles, and updates to references.

Click on the links above to read more information about each release.

Wiki Team

The wiki team is responsible for maintenance of the wiki infrastructure as well as technical review of all materials prior to publication.

- Claus Ballegaard Nielsen, Cranfield University.
- Kimberly Francia, IEEE

The wiki is currently supported by Daniel Robbins of WikiWorks.

SEBoK v. 1.4 released 29 June 2015

SEBoK Discussion

Please provide your comments and feedback on the SEBoK below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account. Simply type your comment in the text field below and DISQUS will guide you through the login or registration steps. Feedback will be archived and used for future updates to the SEBoK. *If you provided a comment that is no longer listed, that comment has been adjudicated. You can view adjudication for comments submitted prior to SEBoK v. 1.0 at SEBoK Review and Adjudication. Later comments are addressed and changes are summarized in the Letter from the Editor and Acknowledgements and Release History.*

If you would like to provide edits on this article, recommend new content, or make comments on the SEBoK as a whole, please see the SEBoK Sandbox [19].

ENCODED_CONTENT
NDc5ODgPGRpdiBpZD0iZGlzcXVzX3RocmVhZCI+PC9kaXY+CjxzY3JpcHQgdHlwZT0idGV4dC9qYXZhc2NyaX
END_ENCODED_CONTENT

Part 7: SE Implementation Examples

Contents

Articles

References

Part 7: SE Implementation Examples

Systems Engineering Implementation Examples

To illustrate the principles described in the Systems Engineering Body of Knowledge (SEBoK) Parts 1-6, Part 7 is a collection of systems engineering (SE) implementation examples. These examples describe the application of SE practices, principles, and concepts in real settings. The intent is to provide typical instances of the application of SE so readers can learn from these experiences. This can improve the practice of SE by illustrating to students, educators, and practitioners the benefits of effective practice, as well as the risks and liabilities of poor practice.

A matrix of implementation examples is used to map the implementation examples to topics in the SEBoK, primarily Part 3. To provide a broader set of domains, both formal case studies and shorter vignettes are used. For the case studies, an introduction and analysis of the case is given with references to the full external case study. For the vignettes, the implementation example is described directly. In the SE literature, a wide variety of examples and formats are considered "case studies." Here, the distinction between a case study and a vignette is that a vignette is a short wiki article written for the SEBoK and a case study exists externally in the SE literature.

An initial set of examples is included with the anticipation that more will be added over time to highlight the different aspects and applications of SE. In addition, new examples can be added to demonstrate the evolving state of practice, such as the application of model-based SE and the engineering of complex, adaptive systems.

Knowledge Areas in Part 7

Each part of the SEBoK is divided into knowledge areas (KAs), which are groupings of information with a related theme. Part 7 is organized into the following KAs:

- Matrix of Implementation Examples
- Case Studies
- Vignettes

References

Works Cited

None.

Primary References

None.

Additional References

None.

< Previous Article | Parent Article | Next Article >

SEBoK v. 1.4 released 29 June 2015

SEBoK Discussion

Please provide your comments and feedback on the SEBoK below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account. Simply type your comment in the text field below and DISQUS will guide you through the login or registration steps. Feedback will be archived and used for future updates to the SEBoK. *If you provided a comment that is no longer listed, that comment has been adjudicated. You can view adjudication for comments submitted prior to SEBoK v. 1.0 at SEBoK Review and Adjudication. Later comments are addressed and changes are summarized in the Letter from the Editor and Acknowledgements and Release History.*

If you would like to provide edits on this article, recommend new content, or make comments on the SEBoK as a whole, please see the SEBoK Sandbox [1].

ENCODED_CONTENT
MTM3NDAPGRpdiBpZD0iZGlzcXVzX3RocmVhZCI+PC9kaXY+CjxzY3JpcHQgdHlwZT0idGV4dC9qYXZhc2Ny
END_ENCODED_CONTENT

References

[1] http://www.sebokwiki.org/sandbox/

Matrix of Implementation Examples

The following matrix maps the SEBoK Systems Engineering Implementation Examples to topics in the Systems Engineering Body of Knowledge (SEBoK), primarily Part 3. It provides both a list of potential systems engineering implementation examples for topics of interest, and a list of relevant topics for each implementation example. Since the number of topics in the SEBoK is extensive, only a subset are included here for clarity. For additional information, see the implementation example of interest and the corresponding SEBoK topic.

Organization and Mapping of Case Studies to the SEBoK

The following short titles shown in Table 1 (Developed for BKCASE) are used for the Case Study implementation examples:

Table 1. Short Titles for the SEBoK Case Studies. (SEBoK Original)

Case Studies	
BT	Business Transformation
ATC	NextGen Air Traffic Control
NASA	NASA's Mission to Saturn
HST	Hubble Space Telescope
GPS	Global Positioning System
GPS II	Global Positioning System II
Radiation	Medical Radiation
FBI VCF	FBI Virtual Case File System
MSTI	Miniature Seeker Technology Integration
Infusion Pump	Next Generation Medical Infusion Pump
DfM	Design for Maintainability

CAS	Complex Adaptive Operating System

Table 2 shows how the topics (each row) align with the Case Study implementation examples (each column):

Table 2. Implementation Examples. Coverage of SEBoK Topics for Each Case Study (SEBoK Original)

SEBoK Topic (Part 3)	BT	ATC	NASA	HST	GPS	GPS II	Radiation	FBI VCF	MSTI	Infusion Pump	DfM	CAS
Systems Thinking					X	X		X	X	X	X	X
Models and Simulation		X	X			X						
Product Systems Engineering				X	X		X	X				
Service Systems Engineering			X			X						
Enterprise Systems Engineering	X	X	X			X						
Systems of Systems (SoS)	X	X				X						
Life Cycle Models	X			X	X				X	X	X	X
Business or Mission Analysis	X	X				X		X		X		
Stakeholder Needs and Requirements	X	X		X		X		X		X		
System Requirements				X	X			X		X		
System Architecture	X	X			X	X			X	X		
System Analysis			X		X							X
System Implementation					X						X	
System Integration	X		X	X	X	X			X		X	
System Verification			X		X		X		X	X		
System Validation	X				X	X	X		X	X		
System Deployment					X	X					X	
Operation of the System					X	X				X	X	
System Maintenance												
Logistics												
Planning	X	X		X				X		X		
Assessment and Control							X	X	X			
Risk Management			X	X	X	X	X	X		X		X
Measurement									X			
Decision Management	X								X			X
Configuration Management					X	X			X		X	X
Information Management			X									X
Quality Management												
Enabling Systems Engineering			X		X	X			X	X	X	X
Related Disciplines						X	X	X		X		X

The following short titles shown in Table 3 (Developed for BKCASE) are used for the Vignette implementation examples:

Table 3. Short Titles for the Vignettes. (SEBoK Original)

Case Studies	
Bag Handling	Denver Airport Baggage Handling System
VA Sub	Virginia Class Submarine
Route Mod	UK West Coast Route Modernisation Project
Water Mgmt	Singapore Water Management
FAA AAS	FAA Advanced Automation System
Light Rail	Standard Korean Light Transit System

Table 4 shows how the topics (each row) align with the Vignette implementation examples (each column):

Table 4. Implementation Examples. Coverage of SEBoK Topics for Each Vignette. (SEBoK Original)

SEBoK Topic (Part 3)	Bag Handling	VA Sub	Route Mod	Water Mgmt	FAA AAS	Light Rail
Business or Mission Analysis			X	X		X
Stakeholder Needs and Requirements			X	X		X
System Requirements			X		X	X
System Architecture	X	X		X	X	X
System Analysis	X			X	X	X
System Implementation						X
System Integration	X			X		X
System Verification	X				X	X
System Validation	X				X	X
System Deployment						X
Operation of the System				X		X
System Maintenance						X
Logistics	X					
Planning	X		X	X	X	X
Assessment and Control					X	X
Risk Management	X		X	X	X	X
Measurement						X
Decision Management	X		X	X		
Configuration Management	X		X		X	
Information Management			X			
Quality Management				X		

References

Works Cited

None.

Primary References

None.

Additional References

None.

< Previous Article | Parent Article | Next Article >

SEBoK v. 1.4 released 29 June 2015

SEBoK Discussion

Please provide your comments and feedback on the SEBoK below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account. Simply type your comment in the text field below and DISQUS will guide you through the login or registration steps. Feedback will be archived and used for future updates to the SEBoK. *If you provided a comment that is no longer listed, that comment has been adjudicated. You can view adjudication for comments submitted prior to SEBoK v. 1.0 at SEBoK Review and Adjudication. Later comments are addressed and changes are summarized in the Letter from the Editor and Acknowledgements and Release History.*

If you would like to provide edits on this article, recommend new content, or make comments on the SEBoK as a whole, please see the SEBoK Sandbox [1].

ENCODED_CONTENT
MTIwODgPGRpdiBpZD0iZGlzcXVzX3RocmVhZCI+PC9kaXY+CjxzY3JpcHQgdHlwZT0idGV4dC9qYXZhc2Nya
END_ENCODED_CONTENT

Case Studies

Systems engineering principles described in the Systems Engineering Body of Knowledge (SEBoK) Parts 1-6 are illustrated in Part 7, Systems Engineering Implementation Examples. These examples describe the application of systems engineering practices, principles, and concepts in real settings. These systems engineering examples can be used to improve the practice of systems engineering by illustrating to students and practitioners the benefits of effective practice and the risks of poor practice. The SEBoK systems engineering implementation examples are grouped in two categories, Case Studies (glossary) and Vignettes (glossary). Case studies reference cases that have already been published by external sources in the existing literature. Vignettes are short wiki articles written specifically for the SEBoK.

List of Case Studies

The following case studies are included:

- Successful Business Transformation within a Russian Information Technology Company
- Federal Aviation Administration Next Generation Air Transportation System
- How Lack of Information Sharing Jeopardized the NASA/ESA Cassini/Huygens Mission to Saturn
- Hubble Space Telescope Case Study
- Global Positioning System Case Study
- Medical Radiation Case Study
- FBI Virtual Case File System Case Study
- MSTI Case Study
- Next Generation Medical Infusion Pump Case Study
- Design for Maintainability
- Complex Adaptive Operating System

Value of Case Studies

Case studies have been used for decades in medicine, law, and business to help students learn fundamentals and to help practitioners improve their practice. A Matrix of Implementation Examples is used to show the alignment of systems engineering case studies to specific areas of the SEBoK. This matrix is intended to provide linkages between each implementation example to the discussion of the systems engineering principles illustrated. The selection of case studies cover a variety of sources, domains, and geographic locations. Both effective and ineffective use of systems engineering principles are illustrated.

The number of publicly available systems engineering case studies is growing. Case studies that highlight the aerospace domain are more prevalent, but there is a growing number of examples beyond this domain.

The United States Air Force Center for Systems Engineering (AF CSE) has developed a set of case studies "to facilitate learning by emphasizing the long-term consequences of the systems engineering/programmatic decisions on cost, schedule, and operational effectiveness." (USAF Center for Systems Engineering 2011) The AF CSE is using these cases to enhance SE curriculum. The cases are structured using the Friedman-Sage framework (Friedman and Sage 2003; Friedman and Sage 2004, 84-96), which decomposes a case into contractor, government, and shared responsibilities in the following nine concept areas:

1. Requirements Definition and Management
2. Systems Architecture Development
3. System/Subsystem Design
4. Verification/Validation
5. Risk Management

6. Systems Integration and Interfaces
7. Life Cycle Support
8. Deployment and Post Deployment
9. System and Program Management

This framework forms the basis of the case study analysis carried out by the AF CSE. Two of these case studies are highlighted in this SEBoK section, the Hubble Space Telescope Case Study and the Global Positioning System Case Study.

The United States National Aeronautics and Space Administration (NASA) has a catalog of more than fifty NASA-related case studies (NASA 2011). These case studies include insights about both program management and systems engineering. Varying in the level of detail, topics addressed, and source organization, these case studies are used to enhance learning at workshops, training, retreats, and conferences. The use of case studies is viewed as important by NASA since "organizational learning takes place when knowledge is shared in usable ways among organizational members. Knowledge is most usable when it is contextual" (NASA 2011). Case study teaching is a method for sharing contextual knowledge to enable reapplication of lessons learned. The MSTI Case Study is from this catalog.

References

Works Cited

Friedman, G.R., and A.P. Sage. 2003. *Systems Engineering Concepts: Illustration Through Case Studies.*

Friedman, G.R., and A.P. Sage. 2004. "Case Studies of Systems Engineering and Management in Systems Acquisition." *Systems Engineering.* 7 (1): 84-96.

NASA. 2011. *A Catalog of NASA-Related Case Studies.* Goddard Space Flight Center: Office of the Chief Knowledge Officer, National Aeronautics and Space Administration (NASA). Updated June 2011. Accessed September 2011. Available: http:/ / www. nasa. gov/ centers/ goddard/ pdf/ 450420main_NASA_Case_Study_Catalog.pdf.

United States Air Force (USAF) Center for Systems Engineering. 2011. *Why Case Studies?*. Wright-Patterson Air Force Base, Ohio, USA: Air Force Institute of Technology (AFIT), US Air Force. Accessed September 2011. Available: http://www.afit.edu/cse/cases.cfm.

Primary References

Friedman, G., and A.P. Sage. 2004. "Case Studies of Systems Engineering and Management in Systems Acquisition". Systems Engineering 7(1): 84-96.

Gorod, A., B.E. White, V. Ireland, S.J. Gandhi, and B.J. Sauser. 2014. *Case Studies in System of Systems, Enterprise Systems, and Complex Systems Engineering*. Boca Raton, FL: CRC Press, Taylor & Francis Group.

NASA. A Catalog of NASA-Related Case Studies. Greenbelt, MD, USA: Office of the Chief Knowledge Officer, Goddard Space Flight Center, National Aeronautics and Space Administration (NASA). Updated June 2011. Accessed December 5 2014 at NASA http:/ / www. nasa. gov/ centers/ goddard/ pdf/ 450420main_NASA_Case_Study_Catalog.pdf.

United States Air Force (USAF) Center for Systems Engineering. 2011. Why Case Studies?. Wright-Patterson Air Force Base, OH, USA: Air Force Institute of Technology (AFIT).

Additional References

None.

< Previous Article | Parent Article | Next Article >

SEBoK v. 1.4 released 29 June 2015

SEBoK Discussion

Please provide your comments and feedback on the SEBoK below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account. Simply type your comment in the text field below and DISQUS will guide you through the login or registration steps. Feedback will be archived and used for future updates to the SEBoK. *If you provided a comment that is no longer listed, that comment has been adjudicated. You can view adjudication for comments submitted prior to SEBoK v. 1.0 at SEBoK Review and Adjudication. Later comments are addressed and changes are summarized in the Letter from the Editor and Acknowledgements and Release History.*

If you would like to provide edits on this article, recommend new content, or make comments on the SEBoK as a whole, please see the SEBoK Sandbox [1].

ENCODED_CONTENT
MjAwMDIPGRpdiBpZD0iZGlzcXVzX3RocmVhZCI+PC9kaXY+CjxzY3JpcHQgdHlwZT0idGV4dC9qYXZhc2Nya
END_ENCODED_CONTENT

Successful Business Transformation within a Russian Information Technology Company

This article describes a successful business transformation of an information technology enterprise. The topic may be of particular interest, especially because this transformation was accomplished by a Russian company during the republic's fast growing economic recovery.

For addition information, refer to the closely related topics of Enabling Businesses and Enterprises and Enterprise Systems Engineering.

Background

In 2001, the top management of the IBS company [1] in Moscow initiated a fundamental transformation to change the company's strategy and business model. The company was one of the biggest Russian information technology (IT) systems integrators at that time, with about 900 employees. Annual revenues of about $80M were mainly generated by information technology (IT) infrastructure projects (complex computing systems, multi-service networks, etc.) and hardware and software distribution. The transformation of the company to form new capabilities in IT services and the associated consulting area is the main topic in the case study.

During the transformation period (from 2001 to the present) IBS was represented as a set of autonomous business units (BUs), called constituent systems, which are virtual, independent businesses with the following characteristics.

- Profit and loss reporting was required for each BU according to management accounting procedures
- BU management established and independently conducted human resources, technology, and product policy
- A centralized back-office was organized to provide supporting functions for each BU. Thus, BUs do not have back-offices; they rely on and “pay” a corporate governing center (CGC) for these services.

A thorough Enterprise System (glossary) (ES) transformation was executed as a set of activities: mission analysis and capabilities decomposition, business architecting, planning of the project program, and implementation of the new business model.

Before and after transformation IBS was an exemplar directed System of Systems (SoS) (glossary): the constituent BUs are autonomous but their operations are supervised by CGC. At the same time IBS also has significant features of an acknowledged SoS: the constituent BUs retain their independent development and sustainment approaches, and changes in the company are based on collaboration between the CGC and each constituent; even operations of BUs are not controlled but only supervised/governed by the CGC through "soft" recommendations and coordination.

IBS was a quite mature ES before the transformation, and it was thoroughly upgraded to form new capabilities of the whole system as well as of the constituents.

Purpose

In 2000-2001 IBS management forecasted considerable growth of the Russian IT services and consulting market based on the fast growing Russian economy, which was rapidly recovering from the national financial crisis of 1998. The largest corporations started overseas expansion and borrowed from international markets to finance this growth. IBS predicted corresponding growth in the complexity of business processes and their associated software and hardware systems all of which should require more consulting and IT services.

Based on this forecast, management established a strategy goal to double the share of IT services and consulting from 25% to 50% over one year; further growth in this business was planned as a long term trend.

The consulting and IT services business is very complex technologically and organizationally and dramatically differs from IBS's former infrastructure focus. Thus, a fundamental transformation was required, and it was executed during 2002.

Initially detected problems appeared as expenditures exceeding resources, slow delivery of the projects and reworking. Later, as it was expected, new problems appeared, for example, disinterest of BUs' managers in developing new technologies or raising qualified employees' motivation. All those problems were solved during transformation and during further development.

The first step of the transformation included strategic analysis and mission-to-capabilities decomposition. Five major capability groups to be focused on were defined. The groups and exemplar capabilities for each group are represented at Figure 1.

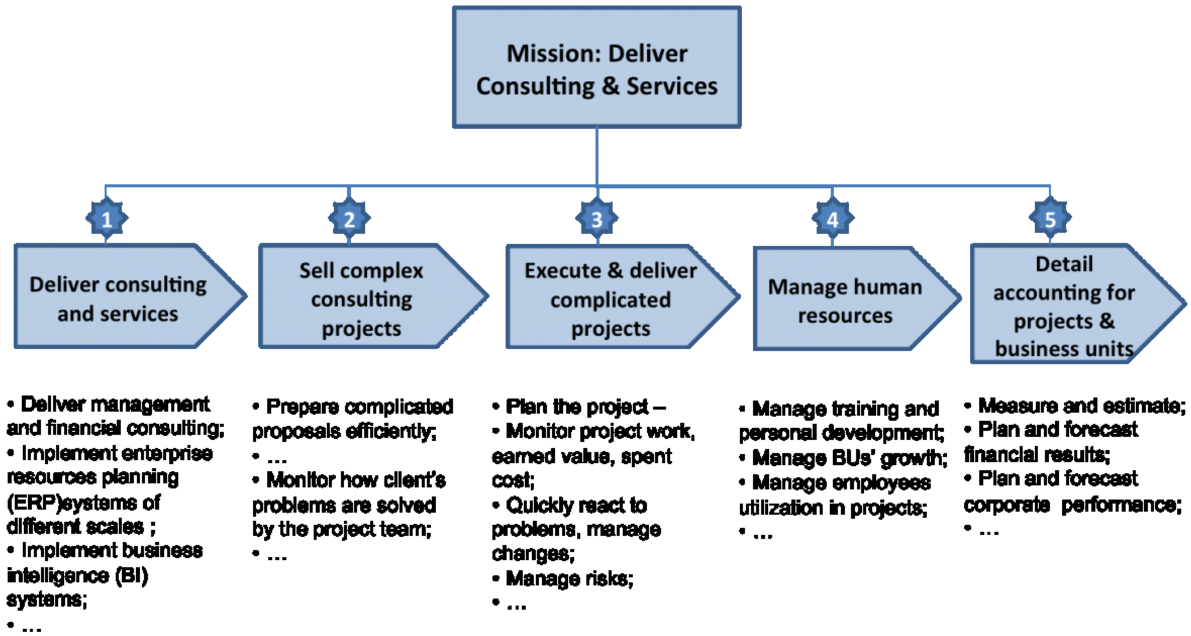

Figure 1. Mission and capabilities desired. (Belov 2014) Reprinted with permission of Taylor and Francis, New York, NY. All other rights are reserved by the copyright owner.

Challenges

All main challenges were caused by knowledge/information deficit described by three factors listed as a, b, and c below.

a. The lack of experience in enterprise transformation (and capability based approaches, even the lack of any textbooks or guides in those areas) was the major challenge which IBS management faced. The task to be solved did not devolve to organizational changes (which was a well-developed and described area), but was appropriately allocated to Enterprise System (glossary) or system of systems (SoS) engineering. In spite of the lack of experience it was decided to prepare and execute the transformation based on the company's employees without involving external consultants. The following arguments supported the decision.

- The task to be solved was not typical, so there weren't widely used and well tested algorithms or methods, and there weren't a lot of consultants experienced in exactly what was needed. So only consultants with general experience (strategy consulting, organizational management) might be hired.
- The Russian consulting industry in 2001-2002 was not well developed, so only foreign professionals were available. But foreign consultants would have needed to study Russian specifics; such study would have unduly lengthened the duration and increased the cost of the transformation.
- A joint transformation team would have to be formed, and IBS employees would have to be involved: management would have to be interviewed and be involved in decision making. In any case all employees would have to participate in change implementations.
- External consultants are not stakeholders; so their level of interest in helping to achieve success might not be very high, and their output also might not be outstanding.
- Unwillingness to open professional secrets and other intellectual property issues to direct competitors were other factors that prevented hiring of external consultants.

Thus, the final decision was to execute the transformation without involvement of external consulting resources. A special BoU responsible for business processes development was established and an agile (glossary) program management approach was applied to handle challenges and to pursue opportunities as well as to mitigate risks.

b. A very high complexity IBS as an enterprise system or SoS. Management recognized that the company and its environment was very complex, with a lot different agents, many constituents, and countless relationships; and that an enterprise system or SoS might become even more complex after transformation. This complexification happened as the company became an “extended enterprise”, the governing hierarchies weakened, and the demand for more sophisticated relationships increased.

c. The risk of mistaken forecast of IT market development. The expected growth of the consulting and services market might have not happened. In this case the transformation would have been senseless. This challenge generated additional emotional stress for management.

Systems Engineering Practices

The SE task of the transformation was established in the form: to develop required capabilities for an enterprise system or SoS – IBS company. The SE process might be represented by the following specific IBS interpretation of the Vee (V) Model (glossary) (“V model”) with Stages 1 through 7 (Figure 2).

Initially (Stage 1) the mission was translated to capabilities (Figure 1); “understanding the constituent systems (BUs) and their relationships” was executed. The transformation team found that capabilities might not be directly translated to any business-agent. Neither BUs (they serves as resource pools), nor projects (being temporal elements), nor employees (each of them have a finite set of skills, experience, responsibilities, etc.) might realize necessary capabilities.

Realizing this (Stage 2) transformation team defined several key areas (Figure 2) of company’s operations or activities which were supposed to be changed to form new capabilities. Appropriate artifacts (procedures, guides, documents, software systems) to support new capabilities were developed and implemented for each of the areas; these new assets formed exactly the corporate infrastructure of new business model.

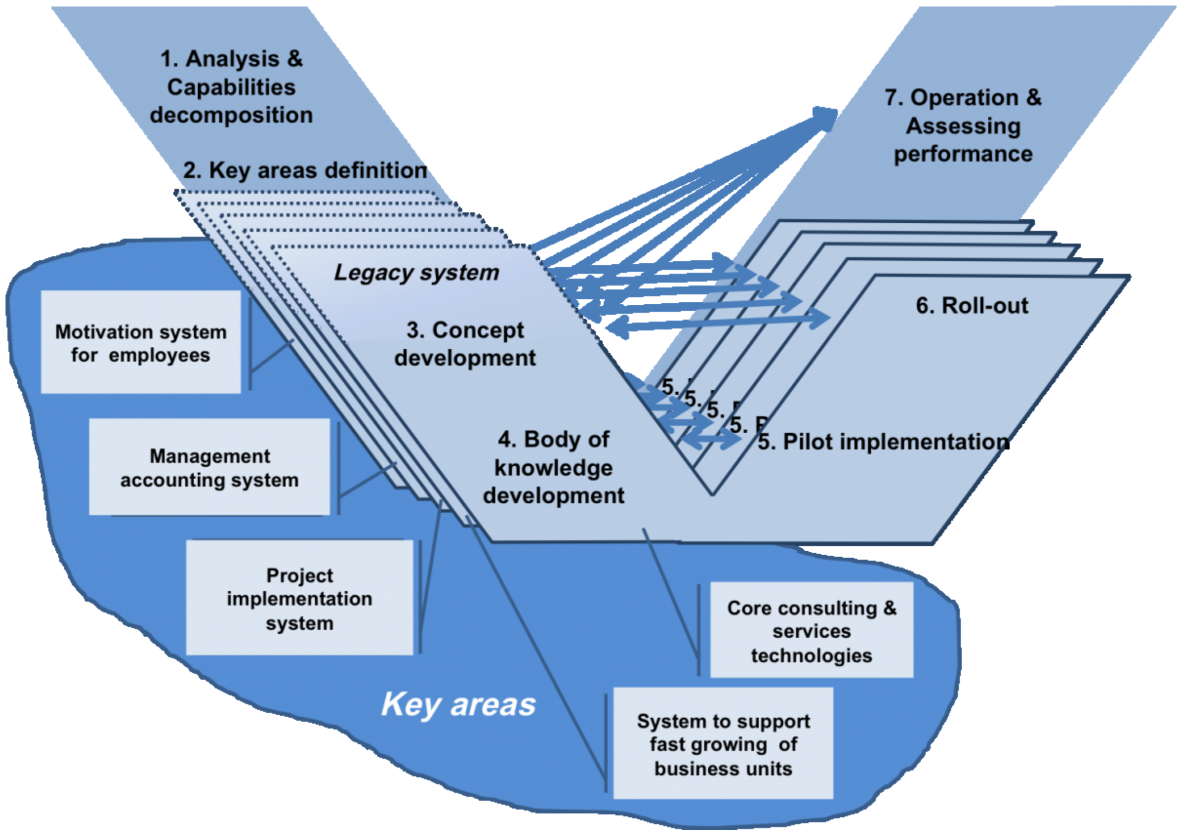

Figure 2. "V model" of systems engineering process of the transformation. (Belov 2014) Reprinted with permission of Taylor and Francis, New York, NY. All other rights are reserved by the copyright owner..

For each new and legacy system (Stage 3) a set of conceptual design documents was developed, describing approaches, polices, processes, and procedures. The entire set of documents formed the business architecture description of the company. The description connected all key areas and defined a target operation model of the company after transformation. This architecture represented multiple views of the IBS company, and thus aptly reflected its enterprise system or SoS nature.

Somewhat in contrast with the conventional linear systems engineering approach advocated by the V model, Stages 4-6 were conducted in parallel to save time and resources. The company's performance (Stage 7) should be monitored based on indicators' measurements, and improvements should be developed and implemented (arrows from Stage 3 to Stage 7). Such iterations have been executed in practice not only during transformation but also later, when procedures, guides and the whole systems were updated.

Integration and interoperability of the new systems required a thorough integration of parallel development jobs. So joint workgroups were formed of the employees at the level of low officers; and CGC played the role of integrated workgroup at the management level. Actually, multi-level integrated workgroups were formed.

The major complexity and risks derived from the challenges described above.

The transformation team developed and used an approach which is very similar to the agile development approach to address those risks. The following principles were used to manage the portfolio of projects in case of uncertainty and deficit of knowledge.

- Form solutions as fast as possible (but not necessarily with pure quality) to test them in practice faster.
- Recognizing failures are unavoidable, perceive them readily and react rationally.
- In case of failure analyze the situation and find a new solution, generate changes, and update the plan.
- Work in parallel, verifying and coordinating intermediate results.

- The schedule might be corrected and updated but should not be jeopardized by improper execution.
- Formulate and test the most critical and most questionable solutions at first.
- Start from pilot area and then expand to embrace the entire scope.
- Use high quality monitoring and a “manual control mode” for piloting and testing developing solutions but not additional aspects to limit waste of the resources.

Following those principles including a very strong discipline of execution, a high level of the sponsorship and all-employee involvement enabled the transformation to be completed on time without hiring consultants while keeping and developing on-going business.

Lessons learned

IBS’s accomplishment of the mission was the major result of ES transformation. Shareholders and management recognized that new capabilities had been formed, that the company could deliver consulting and services, sell and execute complex projects, manage consulting resources effectively, measure its performance, and plan and forecast financial results. Created capabilities are emergent in some sense because they are not directly related to concrete constituents (BUs, or employee, or projects) but are realized by means of integrated end-to-end processes and functions, which are executed in the projects by employees.

The systems organization did not dramatically change during transformation; “visible structure” was not practically changed: no new types of business-agents appeared, existing types did not change much. Those factors did not create new capabilities. Target capabilities were formed as the result of development and implementation of, it would seem, auxiliary and supporting tools – new capabilities support systems. New capabilities were formed mainly by the changes in the intangible areas of governing media, corporate culture, relations, and personnel competences; as well as by the creation of new capabilities support systems; without considerable changes in main company’s business-agents. (Refer to Figure 3.)

The main challenges which management faced (the lack of experience and the ambiguity of market growth forecast) made the uncertainty factor the critical one in the transformation.

What Worked and Why?

An agile program management in general demonstrated its efficiency and applicability to “soft and uncertain” tasks, especially in triggering a pre-established process for dealing with unexpected events; the main aspects of the approach are:

- Senior and credible sponsors
- Multi-level integrated project team(s)
- Open information exchange
- Partnership and collaboration
- Proactive and motivated parties and constituents
- Creative and innovative way of development

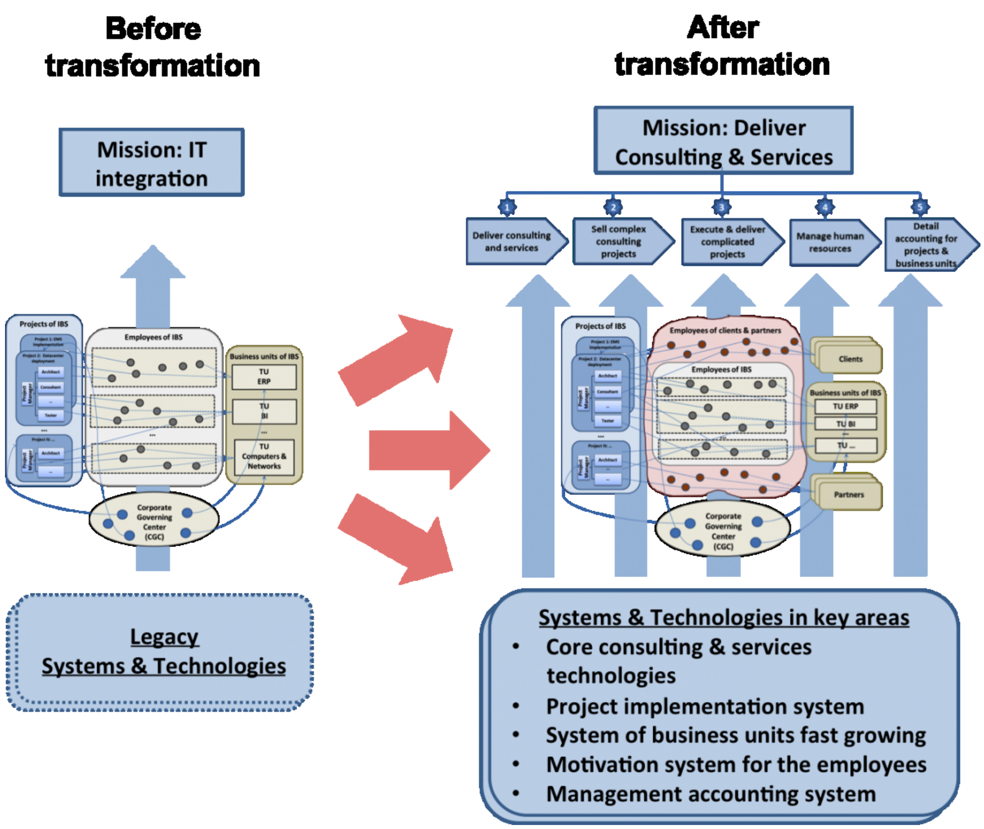

Figure 3. The results of transformation. (Belov 2014) Reprinted with permission of Taylor and Francis, New York, NY. All other rights are reserved by the copyright owner.

- Prioritizing and focusing on the most ambiguous elements of systems design
- Piloting and subsequent roll-out in realistic environments
- Strong project scope control
- Strong project execution control – time schedule and resources control.

What Did Not Work and Why?

Perhaps corporate knowledge base development was the only more or less serious task which was not solved in transformation. The company's management understood the usefulness of knowledge accumulation and further alienation from the carriers in utilizing their business knowledge, so the goal of developing their own knowledge base was established. Special database and software system were developed with appropriate guides, reports and data collection forms; but formal regulation to fill in engineering knowledge accumulation templates did not work. However later this issue progressed quite naturally and simply: common folders were established to store project data in free formats. Such folders served to accumulate knowledge but in flat, unstructured form.

Best Practices and Replication Prospects

The following methods and approaches were proven as efficient and convenient in transformation.

1. Capability based development approach and capability based architecting might be recommended to be utilized in creation and transformation of an enterprise system or SoS. These focused all efforts on the required capabilities and involved very important relations from mission to capabilities and to functions in systems engineering process.
2. An agile program management might be used to solve a wide range of fuzzy and ambiguous problems of different scale in the areas of SE, ES engineering, SoS engineering where there is much uncertainty and lacks of expertise and proven methods or algorithms to solve them. The combination of “soft” and very creative designs with strong planning and progress control is the crucial foundation of this approach.
3. Key area definition and development appropriate to generating new capabilities for support systems (core consulting and services technologies, project implementation systems, systems for business unit growth, management accounting systems, motivation systems). Precisely defining these areas and developing integrated systems in these areas might be considered as quite common for application to a broader group of ESs.

References

Works Cited

Belov, M. 2014. "IBS Group, Eastern European ITS Services – Capability-Based Development for Business Transformation," in *Case Studies in System of Systems, Enterprise Systems, and Complex Systems Engineering*, edited by A. Gorod et al. Boca Raton, FL, USA: CRC Press, Taylor & Francis Group.

Primary References

Belov, M. 2014. “IBS Group, Eastern European ITS Services – Capability-Based Development for Business Transformation.” Case Studies in System of Systems, Enterprise Systems, and Complex Systems Engineering. Gorod, A., B. E. White, V. Ireland, S. J. Gandhi, and B. J. Sauser. Boca Raton, FL: CRC Press, Taylor & Francis Group. Scheduled for publication in July 2014. http://www.taylorandfrancis.com/books/details/9781466502390/
.

Additional References

None

< Previous Article | Parent Article | Next Article>

SEBoK v. 1.4 released 29 June 2015

SEBoK Discussion

Please provide your comments and feedback on the SEBoK below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account. Simply type your comment in the text field below and DISQUS will guide you through the login or registration steps. Feedback will be archived and used for future updates to the SEBoK. *If you provided a comment that is no longer listed, that comment has been adjudicated. You can view adjudication for comments submitted prior to SEBoK v. 1.0 at SEBoK Review and Adjudication. Later comments are addressed and changes are summarized in the Letter from the Editor and Acknowledgements and Release History.*

If you would like to provide edits on this article, recommend new content, or make comments on the SEBoK as a whole, please see the SEBoK Sandbox [1].

ENCODED_CONTENT
Mzg5NzcPGRpdiBpZD0iZGlzcXVzX3RocmVhZCI+PC9kaXY+CjxzY3JpcHQgdHlwZT0idGV4dC9qYXZhc2NyaX
END_ENCODED_CONTENT

References

[1] http://www.en.ibs.ru/

Federal Aviation Administration Next Generation Air Transportation System

This article describes a massive undertaking to modernize the air traffic management enterprise (glossary). The topic may be of particular interest to those involved in air transportation whether in connection with their careers or as pilots or passengers on airplanes. For addition information, refer to the closely related topics of Enabling Businesses and Enterprises and Enterprise Systems Engineering.

Background

This case study presents the Systems Engineering (glossary) and Enterprise Systems Engineering (ESE) (glossary) efforts in the Next Generation (NextGen) Air Transportation Systems by the Federal Aviation Administration (FAA 2008). NextGen is an unprecedented effort by multiple U.S. federal organizations to transform the U.S. air transportation infrastructure (glossary) from a fragmented ground-based navigation system to a net-centric satellite-based navigation system. This project is unique to the FAA because of its large scale, the huge number of stakeholder(s) involved, the properties of the system of interest, and the revolutionary changes required in the U.S. Air Transportation Network (U.S. ATN) enterprise.

A Sociotechnical System (glossary) like the U.S. ATN is a "large-scale [system] in which humans and technical constituents are interacting, adapting, learning, and coevolving. In [such] systems technical constraints and social and behavioral complexity are of essential essence". (Darabi and Mansouri 2014). Therefore, in order to understand changes in the U.S. ATN it was seen as necessary to view it through a lens of evolutionary adaptation rather than rigid systems design. The U.S. ATN serves both military and commercial aircraft with its 19,782 airports, including 547 are commercial airports. Nineteen major airlines, with more than a billion dollars in annual total revenue, along with other 57 national and regional airlines, transport 793 million passengers and realize 53 billion revenue ton-miles.

The Air Traffic Organization (ATO) is responsible for ensuring aircraft navigation in the U.S. National Air Space (NAS) system using a five-layer architecture (glossary). Each aircraft goes through different layers and possibly various zones of this architecture as it takes off from an airport until its lands at another airport (Donohue and Zellweger 2001). However, this architecture is fragmented and many issues are raised: an airplane's path through its route is not optimized, and the path may change its direction from one zone to another, the destination airport's capacity is limited by the current regulations of minimum aircraft separation distance due to navigation limitations, the real-time weather information is not integrated into the system, and communications are mainly voice-based, etc.

In NextGen major changes to the U.S. ATN design are planned. As already stated, the navigation system will be changed from ground-based communication to satellite-based navigation. The current fragmented architecture will be integrated into a seamless net-centric information system in which the digital communication will replace the current voice communications. Moreover, weather information will be assimilated into decision making and planning across the system.

Purpose

The FAA's purpose is "to provide the safest, most efficient aerospace system in the world". Toward this end the NextGen project is aimed at enhancing the U.S.'s leadership position in air transportation.

During the last three decades the demand for air transportation shows exponential growth. In just one decade from 1995 to 2005 this demand showed a 44% percent increase. Therefore, the change in infrastructure was inevitable. Moreover, 9/11 attacks on the U.S. ATN emphasized this need for change. The combination of a requirement for a safer and more secure network and increasing demand was the motivation for President Bush to enact the Vision 100-Century of Aviation Reauthorization Act on 2003. A major part of this Act was to revolutionize the U.S. ATN by means of the NextGen project. The first integration plan of the project was released in 2004, and the project is estimated to continue until 2025.

The demand behavior of the U.S. ATN shows diverse degrees of congestion among airports. Although there are multitudes of airports in the system, the top 35 most congested airports carried more than 60% of the total traffic consistently during the period of 2000 to 2008. Because the growth of the network demand is not proportional, the demand in congested airports will be even higher.

A study by the Joint Planning and Development Office (JPDO) shows that flight delays in the current network will cause $6.5 billion of economic loss until 2015, and $19.6 billion until 2025. By implementing NextGen the delays are estimated to be reduced by 38% until 2020. Moreover, aircraft CO2 emissions are a major part of environmental pollution in crowded cities; these will be reduced by 14 million metric tons by 2020. The current level of jet fuel usage is also a known problem because of increasing fuel prices. The NextGen project will improve fuel usage by 1.4 billion gallons cumulative through 2020.

NextGen is pursuing multiple goals to retain the U.S. leadership in aviation, to expand the U.S. ATN capacity, to continue to ensure safety, to increase environment protection, to help ensure national air defense, all generally helping to increase the nation's security (JPDO 2007a).

Eight general capabilities are defined in conducting this mission: (1) network-enabled information access, (2) performance-based operations and services, (3) weather assimilated into decision making, (4) layered adaptive security, (5) positioning, navigation, and timing (PNT) services, (6) aircraft trajectory-based operations (TBO), (7) equivalent visual operations (EVO), and (8) super-density arrival/departure operations.

To create the desired capabilities, general areas of transformations are defined as air traffic management operations, airport operations and infrastructure services, net-centric infrastructure services, shared situational awareness services, layered and adaptive security services, environmental management services, safety management services, and performance management services. The detailed changes in each area are discussed in Concept of Operations for NextGen (JPDO 2007a).

Challenges

An instructive part of this case study is observing evolution in understanding challenges from initial steps of the project through current efforts for delivering it. As an overall conclusion, the perspective on challenges shifted from technical problems and intra-organizational issues to more enterprise-wide issues.

The NextGen Implementation Plan 2008 discussed the following challenges (FAA 2008):

- performance analysis, to understand and assess operational capabilities
- policy, to balance responsibility between humans and automation, for environmental management processes, and for global harmonization strategies
- acquisition workforce staffing
- environmental planning, to resolve conflicts with local environmental constraints
- security

- transition from current ground-based navigation to automatic dependent surveillance – broadcast (ADS-B) technology.

A more recent report on Targeted NextGen Capabilities for 2025 (JPDO 2011) highlights the effect of the multi-stakeholder nature of the project on raising additional challenges. Achieving Interagency Collaboration is the first issue, which is important in implementing security, safety, policy making, and technological advancement.

Increasing capacity, reducing delay and protecting the environment are the main three promises of the NextGen project. However, reaching the defined high standards is not an easy task. A major part of this challenge is integrating new technologies into legacy systems, aircraft, airports, facilities, and organizations. Airlines and general aviation pilots resist the expense of additional avionics and communications equipment, even though it bolsters the common good of air travel.

Maintaining airports and airspace security requires coherent and harmonious work of multiple U.S. agencies. The core of this challenge is not just changing the technology but also the processes, organizational structures, and enterprises to meet the new requirements of security.

Moreover, the need for greater information sharing in this net-centric environment is a challenge. The current culture of limited information sharing in which inter-organizational and intra-organizational information is strictly divided creates tension in a seamless information sharing infrastructure. In addition to that, the responsibility of generating, sharing, and utilizing useful information should be addressed in advance to avoid costly mistakes.

Verification and validation of NextGen deliverables is a major issue. The traditional systems engineering methods of verification and validation are tailored for testing an isolated system, while by definition a project like NextGen requires new methodologies of verification and validation beyond the scope of one system. The knowledge and experience of advancement in systems engineering in this area can be of priceless value for future projects.

Balance between human decision-making and automation is required to ensure a correct policy for increasing traffic and safety concerns. Changes in both human resource and technological facilities are required to effectively address this challenge.

The support of local communities is essential to facilitate development of the U.S. ATN and its physical infrastructure.

Communication, navigation, and surveillance systems in NextGen are going through major changes in terms of capacity and technology. However, planning required backups for them in case of any emergency is an area of challenge in developing NextGen.

The rise of Unmanned Aircraft Systems (UASs) provides significant opportunities for both military and commercial applications. However, integrating them into the NAS and developing policing and strategies for safe and secure use is a concern for the revolutionized U.S. ATN.

And finally realizing the benefits of NextGen is dependent on the critical mass of early adopters, similar to any technological advancement. Therefore, the NextGen project authority requires well-defined policies for motivating stakeholders' participation.

Systems Engineering Practices

The FAA NextGen is not just a revolution of the U.S. air transportation infrastructure, but also a shift in its enterprise. The enterprise architecture document, which is developed by JPDO, provides an overview of the desired capabilities (JPDO 2007b).

The Enterprise Architecture (glossary) is described using Department of Defense Architecture Framework (DoDAF) and the Federal Enterprise Architecture (FEA). DoDAF is used to describe the operational aspects of the project. The three views of DoDAF, the Operational View (OV), the Systems View (SV), and the Technical Standards View (TV), are presented in the enterprise architecture document. The Overview and Summary Information (AV-1) is the formal statement about how to use the architecture, the Integrated Dictionary (AV-2) defines the terms in the document, the Community Model (OV-1) presents a high level depiction of the NextGen community, the Operational Node Connectivity Description (OV-2) presents the information flow among operational nodes in the system, Operational Information Exchange Matrix (OV-3) details the description of information flow in OV-2. Other architectural views of the system based on DoDAF are the Activity Model (OV-5) which documents activities (functions and processes), the Operational Event/Trace Description (OV-6c) is a part of sequence and timing description of activities, the System Functionality Description (SV-4) explains system functional hierarchies, and the Operational Activity to System Functionality Traceability Matrix (SV-5) is specification of relationships between operational activities in architecture and functional activities. However, a challenging part of applying this Enterprise Architecture is transformation from legacy systems to the new NextGen. This transformation is the ultimate test for relevance and comprehensiveness of the developed Enterprise Architecture.

Acquisition is the heart of systems engineering activities in the FAA NextGen project. As mentioned in Challenges above, the current practice of verification of validation in systems engineering (SE) is geared toward single isolated systems, rather than a myriad of interconnected System of Systems (SoS) (glossary). Moreover, the capabilities of NextGen are interdependent, and different programs rely on each other to deliver the promises. 250 unique and highly interconnected acquisition programs are identified in the FAA's Capital Investment Plan, and these are to be delivered by 1820 FAA acquisition professionals. In addition, program complexity, budget uncertainty, and the challenge of finding acquisition professionals present other problems. The experience of systems acquisition in NextGen can provide a useful knowledge for future similar projects.

Lessons Learned

Although major portions of the FAA NextGen project are technical transformations and physical infrastructure developments, the transformation in the aviation enterprise is important but to some degree neglected. Part of the issue might be the fact that this transformation is beyond the responsibility and capability of FAA. However, to accomplish NextGen's perceived benefits it is important to realize the effects of legacy systems, and most importantly the legacy enterprise architecture of the U.S. ATN. Many of the actual challenges in the system arose because of this inattention.

The sequestration in the U.S. government, the Budget Control Act of 2011, has cut the project funding substantially in recent years. As a result the project schedule and portfolio are subject to constant and wide-spread changes. The FAA was focused on delivering Optimization of Airspace and Procedures in the Metroplex (OPAM) program which is designed to reduce the delay, fuel consumption, and exhaust emission in busiest airports. The three areas of Houston, North Texas, and Washington D.C. were planned to complete the design phase on 2013 and start implementation.

Out of 700 planned ADS-B ground stations, 445 were operational on February 2013. ADS-B capability is a NextGen descendant of current radar systems and provides situational awareness for the players in the NAS using the Global Positioning System (GPS) and Wide Area Augmentation System (WAAS).

On the enterprise part of the project, the FAA Modernization and Reform Act of 2012 provided financial incentives for airlines and commercial aviation manufacturers to implement the required equipment in their aircraft. These incentives are designed to engage the air transportation community in the project and to create the critical mass of equipped airplanes.

There are considerable practices in applying NextGen. Establishment of the JPDO made the efforts of the project more coherent and integrated. JPDO's main responsibility is to coordinate development of NextGen. The role of this organization is to represent multiple stakeholders of the project, which enables it to resolve possible conflicts of interests inside one entity. Moreover, such an organization provides a venue for technical knowledge-sharing, creating a consensus, and making an integrated system.

Emphasizing delivery of the mid-term objectives of NextGen is another lesson of the project. It was a well-known practice documented by Forman and Maier to establish mid-points for complex projects (Forman 2000). Developing a mid-level system provides the system designers an opportunity to examine their underlying assumptions, to identify best practices and heuristics in the context of the project, and to reapply the acquired knowledge thorough evolutionary developments. A major shift in the policy of FAA in recent years was to focus on delivering project mid-term objectives.

There are unique characteristics of NextGen which makes it a valuable case for learning and replicating to other complex transformation projects of sociotechnical systems. The scale of the project for infrastructure transformation is unprecedented. The system includes legacy systems and cutting edge technology, and its performance is based on their coherent work. The project implementation is dependent on involved participation of multiple governmental and commercial organizations. Moreover, this case-study provides a great investigation in enterprise governance and enterprise transformation beyond a single organization.

References

Works Cited

Darabi, H.R., and M. Mansouri. 2014. "NextGen: Enterprise Transformation of the United States Air Transport Network," in *Case Studies in System of Systems, Enterprise Systems, and Complex Systems Engineering*, edited by A. Gorod et al. Boca Raton, FL, USA: CRC Press, Taylor & Francis Group.

Donohue, G.L., and A.G. Zellweger. 2001. "Air Transportation Systems Engineering." *American Institute of Aeronautics and Astronautics*. 193 (1).

FAA. 2008. *FAA's NextGen Implementation Plan. Federal Aviation Administration*. Washington, DC, USA. Accessed March 29, 2014. Available: http://www.faa.gov/nextgen/media/ng2008_implementation_plan.pdf.

Forman, B. 2000. "The political process and systems architecting," in *The Art of Systems Architecting*, 2nd ed., edited by M.W. Maier and E. Rechtin. Boca Raton, FL, USA: CRC Press LLC.

JPDO. 2007a. *Concept of operations for the next generation air transportation system.* Joint Planning Development Office. Accessed March 29, 2014. Available: http://www.jpdo.gov/library/nextgen_v2.0.pdf.

JPDO. 2007b. *Enterprise Architecture V2.0 for the Next Generation Air Transportation System.* Joint Planning and Development Office. Accessed March 29, 2014. Available: http://www.jpdo.gov/library/EnterpriseArchitectureV2.zip.

JPDO. 2011. *Targeted NextGen Capabilities for 2025*. Joint Planning and Development Office. Accessed March 29, 2014. Available: http://www.jpdo.gov/library/2011_Targeted_NextGen-Capabilities_for_2025_v3.25.pdf.

Primary References

Darabi, H.R. and M. Mansouri. 2014. "NextGen: Enterprise Transformation of the United States Air Transport Network." Case Studies in System of Systems, Enterprise Systems, and Complex Systems Engineering. Gorod, A., B.E. White, V. Ireland, S. J. Gandhi, and B.J. Sauser. Boca Raton, FL: CRC Press, Taylor & Francis Group. Scheduled for publication in July 2014. http://www.taylorandfrancis.com/books/details/9781466502390/. Accessed 29 March 2014.

Additional References

None.

< Previous Article | Parent Article | Next Article>

SEBoK v. 1.4 released 29 June 2015

SEBoK Discussion

Please provide your comments and feedback on the SEBoK below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account. Simply type your comment in the text field below and DISQUS will guide you through the login or registration steps. Feedback will be archived and used for future updates to the SEBoK. *If you provided a comment that is no longer listed, that comment has been adjudicated. You can view adjudication for comments submitted prior to SEBoK v. 1.0 at SEBoK Review and Adjudication. Later comments are addressed and changes are summarized in the Letter from the Editor and Acknowledgements and Release History.*

If you would like to provide edits on this article, recommend new content, or make comments on the SEBoK as a whole, please see the SEBoK Sandbox [1].

ENCODED_CONTENT
MTgxMDYPGRpdiBpZD0iZGlzcXVzX3RocmVhZCI+PC9kaXY+CjxzY3JpcHQgdHlwZT0idGV4dC9qYXZhc2Ny
END_ENCODED_CONTENT

How Lack of Information Sharing Jeopardized the NASA/ESA Cassini/Huygens Mission to Saturn

This article describes a deep space mission where more forthright information exchanges between teamed but rival agencies could have preserved the original plan as well as saved much time and money. The topic may be of particular interest to those involved in institutional collaborations where there are vested interests in protecting rather than sharing information.

For addition information, refer to the closely related topics of Information Management, Organizing Business and Enterprises to Perform Systems Engineering and Fundamentals of Services.

Background

Before the “Faster, Better, Cheaper” philosophy introduced in the 1990s, the United States National Aeronautics and Space Administration (NASA) focused on three classes of unmanned space missions. In order of increasing cost, these were the Discovery, New Frontiers, and Flagship programs. Flagship programs typically cost more than $1B, and included the Voyager (outer planets), Galileo (Jupiter), Cassini-Huygens (Saturn), Mars Science Laboratory (Mars), and the James Webb Space Telescope. (Wall 2012)

The concept of the Cassini-Huygens mission was initiated in 1982 as the result of a working group formed by the National Academy of Sciences and the European Science Foundation. This group sought opportunities for joint space missions; several subsequent reports endorsed the working group’s concept of a Saturn orbiter coupled with a Titan (Saturn’s largest moon) lander. (Russell 2003, p. 61)

By 1988, NASA was politically motivated to reverse earlier tensions with the European Space Agency (ESA) by engaging in a joint mission. Cassini-Huygens was seen as a mechanism to achieve this goal, and the cooperation between NASA and ESA helped the program survive potential budget cuts (since the U.S. was obligated to match ESA commitments). (Russell 2003, p. 62)

NASA and ESA approved the Cassini-Huygens program, and it proceeded under a traditional management approach. NASA built the Cassini orbiter (the largest and most complex unmanned space probe ever built) and the ESA constructed the Huygens lander. This partition of responsibility almost led to the failure (glossary) of the Titan survey portion of the mission. Cassini (which would conduct a variety of scientific surveys of the Saturn planetary system) was expected to relay transmissions from Huygens to NASA's Deep Space Network (DSN); however, the interface between the lander and orbiter was not well-managed and erroneous assumptions about how the orbiter/lander system would behave after separation nearly doomed the Titan exploration portion of the mission. (Oberg 2004)

Purpose

The intent of the Titan survey portion of the Cassini-Huygens mission was that the Huygens lander would separate from the Cassini orbiter and commence a one-way, 2.5 hour descent into Titan's atmosphere. Its modest transmitter would send data back to the orbiter, which would relay the information to Earth. (Oberg 2004, p. 30) This effectively made the radio link between the two spacecraft a single point of failure (SPOF) and one that was not well characterized.

Alenia Spazio SpA, the Italian communications vendor that built the radio system, overlooked the Doppler shift (approximately 38 kHz) (Oberg, 2004, p. 31) that would occur when Huygens separated from Cassini and began its descent (Oberg 2004, p. 38). The communications protocol was binary phase-key shifting: "[the] transmission system represents 1s and 0s by varying the phase of the outgoing carrier wave. Recovering these bits requires precise timing: in simple terms, Cassini's receiver is designed to break the incoming signal into 8192 chunks every second. It determines the phase of each chunk compared with an unmodulated wave and outputs a 0 or a 1 accordingly". (Oberg 2004, p. 31) The receiver was appropriately configured to compensate for the Doppler shift of the carrier wave but would be unable to adjust for the Doppler shift of the encoded data. "In effect, the shift would push the signal out of synch with the timing scheme used to recover data from the phase-modulated carrier." (Oberg 2004, p. 33) Therefore, the communications system would be unable to decode the data from the lander and would then relay scrambled information to NASA. Because of the failure mechanism involved, the data would be completely unrecoverable.

Both Cassini and Huygens had been tested before launch; however, none of the testing accurately reflected the Doppler shift that would be experienced at this critical phase of the mission. An opportunity to conduct a full-scale, high-fidelity radio test was ignored due to budget constraints; the testing would have required disassembly and subsequent recertification of the probes. (Oberg, 2004, p. 30) Correcting this latent issue would have been trivial before the spacecraft were launched (via a minor firmware upgrade); (Oberg 2004, p. 33) once they were on the way to Saturn any corrective action would be severely limited and expensive.

Once the mission was underway, the probe coasted along its seven-year trajectory to Saturn and its moons. Claudio Sollazzo, the ESA ground operations manager, was uncomfortable with the untested communications system. He tasked Boris Smeds, an engineer with radio and telemetry experience, with finding a way to test the communications system using an Earth-generated signal. (Oberg 2004, p. 30)

Smeds spent six months developing the test protocols that would use Jet Propulsion Laboratory (JPL) ground stations and an exact duplicate of Huygens. Simulated telemetry would be broadcast from Earth to Cassini and relayed back; the test signal would vary in power level and Doppler shift to fully exercise the communications link and accurately reflect the anticipated parameters during Huygens's descent into Titan's atmosphere. (ESA 2005)

Challenges

Smeds faced opposition to his test plans from those who felt it was unnecessary, but ultimately prevailed due to support from Sollazzo and Jean-Pierre Lebreton, the Huygens project (glossary) scientist. More than two years after the mission was launched, Smeds traveled to a DSN site in California to conduct the test. (Oberg 2004, p. 31)

A test signal was broadcast, received by Cassini, re-transmitted to the DSN site, and relayed to ESA's European Space Operation Centre (ESOC) in Darmstadt, Germany for analysis. Testing had to be conducted when the orbiter was in the correct relative position in the sky; it was more than a quarter of a million miles away with a signal round-trip time of nearly an hour. The test immediately exposed an issue; the data stream was intermittently corrupted, with failures not correlated to the power level of the test signal. The first of two days of testing concluded with no clear root cause identified. (Oberg 2004, p. 31)

Even though the probe was far from its ultimate destination, many science teams were competing for time to communicate with it using the limited bandwidth available. The communications team would not be able to conduct another set of trials for several months. Smeds diagnosed the root cause of the problem; he felt it was the Doppler shifts induced in the simulated signal. However, the test plan did not include unshifted telemetry (an ironic oversight). He modified his test plan overnight and shortened the planned tests by 60%; this recovered sufficient time for him to inject an unshifted signal into the test protocols. (Oberg 2004, p. 32)

This unshifted signal did not suffer from the same degradation; however, other engineers resisted the diagnosis of the problem. Follow-up testing using probe mockups and other equipment ultimately convinced the ESA of the issue; this took an additional seven months. (Oberg 2004, p. 33)

By late 2000, ESA informed NASA of the latent failure of the communications link between Cassini and Huygens. Inquiry boards confirmed that Alenia Spazio had reused timing features of a communications system used on Earth-orbiting satellites (which did not have to compensate for Doppler shifts of this magnitude). (Oberg, 2004, p. 33) In addition, because NASA was considered a competitor, full specifications for the communications modules were not shared with JPL. The implementation of the communications protocols was in the system's firmware; trivial to correct before launch, impossible to correct after. (ESA 2005)

A 40-man Huygens Recovery Task Force (HRTF) was created in early 2001 to investigate potential mitigation actions. Analysis showed that no amount of modification to the signal would prevent degradation; the team (glossary) ultimately proposed changing the trajectory of Cassini to reduce the Doppler shift. (ESA 2005) Multiple studies were conducted to verify the efficacy of this remedy, and it ultimately allowed the mission to successfully complete the Titan survey.

Systems Engineering Practices

Space missions are particularly challenging; once the spacecraft is en route to its destination, it is completely isolated. No additional resources can be provided and repair (particularly for unmanned mission) can be impossible. Apollo 13's crew barely survived the notable mishap on its mission because of the resources of the docked Lunar Excursion Module (LEM) and the resourcefulness of the ground control team's experts. A less well-known failure occurred during the Galileo mission to Jupiter. After the Challenger disaster, NASA adopted safety standards that restricted the size of boosters carried in the Space Shuttle. (Renzetti 1995) Galileo was delayed while the Shuttles were grounded and Galileo's trajectory was re-planned to include a Venus fly-by to accelerate and compensate for a smaller booster. Galileo's main antenna failed to deploy; lubricant had evaporated during the extended unplanned storage (Evans 2003) and limited computer space led to the deletion of the antenna motor-reversing software to make room for thermal protection routines. When the antenna partially deployed, it was stuck in place with no way to re-furl and redeploy it. Engineers ultimately used an onboard tape recorder, revised transmission protocols, the available low-gain antenna, and ground-based upgrades to the DSN to save the mission. (Taylor, Cheung, and Seo 2002)

The Titan survey was ultimately successful because simulation (glossary) techniques were able to verify the planned trajectory modifications and sufficient reaction mass was available to complete the necessary maneuvers. In addition, Smeds's analysis gave the mission team the time it needed to fully diagnose the problem and develop and implement the remedy. If this test were conducted the day before the survey it would merely have given NASA and ESA advance warning of a disaster. The time provided enabled the mission planners to craft a trajectory that resolved the communication issue and then blended back into the original mission profile to preserve the balance of the Saturn fly-bys planned for Cassini. (Oberg 2004, p. 33)

Lessons Learned

The near-failure of the Cassini-Huygens survey of Titan was averted because a handful of dedicated systems engineers fought for and conducted relevant testing, exposed a latent defect, and did so early enough in the mission to allow for a recovery (glossary) plan to be developed and executed. Root causes of the issue included politically-driven partitioning, poor interface management, overlooked contextual information, and a lack of appreciation for single-points-of-failure (SPOFs).

The desire to use a joint space mission as a mechanism for bringing NASA and ESA closer together (with the associated positive impact in foreign relations) introduced an unnecessary interface into the system. Interfaces must always be managed carefully; interfaces between organizations (particularly those that cross organizational or political borders) require extra effort and attention. Boeing and Airbus experienced similar issues during the development of the Boeing 787 and A380; international interfaces in the design (glossary) activities and supply chains led to issues:

> *…every interface in nature has a surface energy. Creating a new surface (e.g., by cutting a block of steel into two pieces) consumes energy that is then bound up in that surface (or interface). Interfaces in human systems (or organizations), a critical aspect of complex systems such as these, also have costs in the effort to create and maintain them. Second, friction reduces performance. Carl von Clausewitz, the noted military strategist, defined friction as the disparity between the ideal performance of units, organizations, or systems, and their actual performance in real-world scenarios. One of the primary causes of friction is ambiguous or unclear information. Partitioning any system introduces friction at the interface.* (Vinarcik 2014, p. 697)

Alenia Spazio SpA's unclear understanding of the Doppler shift introduced by the planned relative trajectories of Huygens and Cassini during the Titan survey led it to reuse a component from Earth-orbiting satellites. Because it considered NASA a competitor and cloaked details of the communications system behind a veil of propriety, it prevented detection of this flaw in the design phase. (Oberg 2004, p. 33)

Because NASA and ESA did not identify this communication link as a critical SPOF, they both sacrificed pre-launch testing on the altar of expediency and cost-savings. This prevented detection and correction of the flaw before the mission was dispatched to Saturn. The resource cost of the later analysis and remedial action was non-trivial and if sufficient time and reaction mass had not been available the mission would have been compromised. It should be noted that a number of recent spacecraft failures are directly attributable to SPOFs (notably, the Mars Polar Lander (JPL 2000) and the Genesis sample return mission (GENESIS, 2005)). Effective SPOF detection and remediation must be a priority for any product development effort. More generally, early in the development process, significant emphasis should be placed on analyses focused on what might go wrong ("rainy day scenarios") in addition to what is expected to go right ("sunny day scenarios").

The success of the Huygens survey of Titan was built upon the foundation established by Boris Smeds by identifying the root cause of the design flaws in a critical communications link. This case study underscores the need for clear contextual understanding, robust interface management, representative testing, and proper characterization and management of SPOFs.

References

Works Cited

Evans, B. 2003. "The Galileo Trials." Spaceflight Now. Available: http://www.spaceflightnow.com/galileo/030921galileohistory.html.

GENESIS. 2005. "GENESIS Mishap Investigation Board Report Volume I." Washington, DC, USA: National Aeronautics and Space Administration (NASA).

JPL. 2000. "Report on the Loss of the Mars Polar Lander and Deep Space 2 Missions." Special Review Board. Pasadena, CA, USA: NASA Jet Propulsion Laboratory (JPL).

ESA. 2005. "Modest Hero Sparks Team Response." European Space Agency. Available: http://www.esa.int/Our_Activities/Operations/Modest_hero_sparks_team_response.

Oberg, J. 2004. "Titan Calling: How a Swedish engineer saved a once-in-a-lifetime mission to Saturn's mysterious moon." *IEEE Spectrum.* 1 October 2004, pp. 28-33.

Renzetti, D.N. 1995. "Advanced Systems Program and the Galileo Mission to Jupiter." The Evolution of Technology in the Deep Space Network: A History of the Advanced Systems Program. Available: http://deepspace.jpl.nasa.gov/technology/95_20/gll_case_study.html.

Russell, C. 2003. *The Cassini-Huygens Mission: Volume 1: Overview, Objectives and Huygens Instrumentarium*. Norwell, MA, USA: Kluwer Academic Publishers.

Taylor, J., K.-M. Cheung, and D. Seo. 2002. "Galileo Telecommunications. Article 5." *DESCANSO Design and Performance Summary Series*. Pasadena, CA, USA: NASA/Jet Propulsion Laboratory.

Vinarcik, M.J. 2014. "Airbus A380 and Boeing 787 — Contrast of Competing Architectures for Air Transportation," in *Case Studies in System of Systems, Enterprise Systems, and Complex Systems Engineering*, edited by A. Gorod et al. Boca Raton, FL, USA: CRC Press. p. 687-701.

Wall, M. 2012. "NASA Shelves Ambitious Flagship Missions to Other Planets." Space News. Available: http://www.space.com/14576-nasa-planetary-science-flagship-missions.html.

Primary References

None.

Additional References

None.

< Previous Article | Parent Article | Next Article>

SEBoK v. 1.4 released 29 June 2015

SEBoK Discussion

Please provide your comments and feedback on the SEBoK below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account. Simply type your comment in the text field below and DISQUS will guide you through the login or registration steps. Feedback will be archived and used for future updates to the SEBoK. *If you provided a comment that is no longer listed, that comment has been adjudicated. You can view adjudication for comments submitted prior to SEBoK v. 1.0 at SEBoK Review and Adjudication. Later comments are addressed and changes are summarized in the Letter from the Editor and Acknowledgements and Release History.*

If you would like to provide edits on this article, recommend new content, or make comments on the SEBoK as a whole, please see the SEBoK Sandbox [1].

ENCODED_CONTENT
NjYyNTUPGRpdiBpZD0iZGlzcXVzX3RocmVhZCI+PC9kaXY+CjxzY3JpcHQgdHlwZT0idGV4dC9qYXZhc2NyaX
END_ENCODED_CONTENT

Hubble Space Telescope Case Study

This article describes a remarkable engineering feat with vast scientific benefits and implications. The topic may be of particular interest to those facing formidable Systems Engineering (glossary) challenges where one might thrive on a thoughtful blend of humility and optimism. For addition information, refer to the links provided in Section V. Lessons Learned below.

Background

The Hubble Space Telescope (HST) Case Study was developed by the United States Air Force Center for Systems Engineering (AF CSE) located at the Air Force Institute of Technology (AFIT). The AF CSE was tasked to develop case studies focusing on the application of Systems Engineering (glossary) principle (glossary) s within various aerospace program (glossary) s. The HST study (Mattice 2005) is one of four initial case studies selected by AFIT for development in support of systems engineering graduate school instruction. The cases are structured using the Friedman-Sage framework (glossary) (Friedman and Sage 2003; Friedman and Sage 2004, 84-96), which decomposes a case into contractor, government, and shared responsibilities in the following nine concept areas:

1. Requirements Definition and Management
2. Systems Architecture Development
3. System/Subsystem Design
4. Verification/Validation
5. Risk Management
6. Systems Integration and Interfaces
7. Life Cycle Support
8. Deployment and Post Deployment
9. System and Program Management

The case study provides a useful example of the rising cost (glossary) of defect correction through successive life cycle phases, demonstrating how an error (in test fixture specification) that could have been fixed for $1,000 at the design (glossary) stage, or detected and fixed with a $10 million investment in an end-to-end test of the telescope on the ground, ended up costing $1 billion to fix when the system was in service (glossary).

Purpose

The Hubble Space Telescope (HST) is an orbiting astronomical observatory operating in the spectrum from the near-infrared into the ultraviolet. Launched in 1990 and still operational, HST carries and has carried a wide variety of instruments producing imaging, spectrographic, astrometric, and photometric data through both pointed and parallel observing programs. Over 100,000 observations of more than 20,000 targets have been produced for retrieval. The telescope is well known as a marvel of science. This case study hopes to represent the facet of the HST that is a marvel of systems engineering, which, in fact, generated the scientific research and observation capabilities now appreciated worldwide.

Viewed with the clarity that only time and hindsight provide, the HST program certainly represents one of the most successful modern human endeavors on any scale of international scope and complex (glossary) ity. As a systems engineering project the HST had to respond to requirement (glossary) s from the diverse international scientific community at a time when NASA was implementing a different research-development-acquisition philosophy and process (glossary) than what was predominately being using in most major government acquisition programs. As with most other large programs, powerful influences outside the systems engineering process itself became issues that HST Systems Engineer (glossary) s in effect had to acknowledge as integral to their overall system/program/engineering management responsibility.

Challenges

The story of how this remarkable capability came to be is a story of the complicated interactions of a systems engineering process, which we like to believe we understand, with equally demanding political, budgetary, and institutional processes we often fail to understand or comprehend at the time they occur. In the final analysis, these processes are inseparable and integral to attaining program success. The challenge to modern systems engineers is to fully embrace the discipline of the systems engineering process while at the same time learning how to continue to practice it in spite of inevitable external influences and instabilities that often cannot be anticipated.

Major differences revolved around the nature and needs of a very different HST “customer” or user from most DoD systems. The HST had to respond to requirements from the diverse international scientific community instead of from DoD’s combatant commands. In addition, at the time, NASA implemented a different research-development-acquisition philosophy and process than the DoD Acquisition Management Framework described in the DoD 5000 series acquisition reforms. As with most other large programs, powerful influences outside the systems engineering process itself became issues that HST systems engineers in effect had to acknowledge as integral to their overall system/program/engineering management responsibility.

Systems Engineering Practices

During the critical systems engineering phase for the HST program (1970s concept studies thru 1990 launch) there appears to have been no NASA systems engineering master process. Rather, field center processes were operative and possibly even in competition, as centers (especially Marshall and Goddard for HST) were in keen competition for lead management roles and responsibilities. We will see the systems engineering and program management impacts of this competition as it played out for HST, with the science mission objectives and instrumentation payloads being the motivation for Goddard vs. the vehicle/payload access to space motivation of Marshall. In the final analysis, the roles of the major contractors in engineering the system with uneven NASA participation over the system life cycle had a telling effect.

Lessons Learned

Five learning principles (LPs) were derived that address the more broadly applicable areas of systems engineering knowledge. These five LPs inform the areas of the SEBoK that are most strongly related to the case study. The five areas are:

- stakeholder requirements definition (LP1);
- planning (pre-program trade studies) (LP2);
- system integration (LP3);
- life cycle model management (LP4); and
- risk management (LP5).

A synopsis of the HST Learning Principles (LPs) are as follows:

Stakeholder Requirements Definition LP1: Early and full participation by the customer/user throughout the program is essential to success. In the early stages of the HST program the mechanism for involving the customer was not well defined. The user community was initially polarized and not effectively engaged in program definition and advocacy. This eventually changed for the better, albeit driven heavily by external political and related national program initiatives. Ultimately, institutionalization of the user's process for involvement ensured powerful representation and a fundamental stake and role in both establishing and managing program requirements. Over time, the effectiveness of "The Institute" led to equally effective user involvement in the deployment and on-orbit operations of the system as well.

Planning LP 2: The use of Pre-Program Trade Studies (e.g. "Phased Studies" or "Phased Project Planning") to broadly explore technical concepts and alternatives is essential and provides for a healthy variety of inputs from a variety of contractors and government (NASA) centers. These activities cover a range of feasibility, conceptual, alternative and preliminary design trades, with cost initially a minor (later a major) factor. In the case of HST, several NASA Headquarters and Center organizations funded these studies and sponsored technical workshops for HST concepts. This approach can promote healthy or unhealthy competition, especially when roles and responsibilities within and between the participating management centers have not yet been decided and competing external organizations use these studies to further both technical and political agendas. NASA Center roles and missions can also be at stake depending on political and or budgetary realities. The systems engineering challenge at this stage is to "keep it technical, stupid!"

Systems Integration LP 3: A high degree of systems integration to assemble, test, deploy, and operate the system is essential to success and must be identified as a fundamental program resource need as part of the program baseline. For HST, the early wedding of the program to the Shuttle, prior NASA and NASA contractor experience with similarly complex programs, such as Apollo, and the early requirement for manned, on-orbit servicing made it hard not to recognize this was a big systems engineering integration challenge. Nonetheless, collaboration between government engineers, contractor engineers, as well as customers, must be well defined and exercised early on to overcome inevitable integration challenges and unforeseen events.

Life Cycle Models LP 4: Life Cycle Support planning and execution must be integral from day one, including concept and design phases. The results will speak for themselves. Programs structured with real life cycle performance as a design driver will be capable of performing in-service better, and will be capable of dealing with unforeseen events (even usage in unanticipated missions). HST probably represents a benchmark for building in system sustainment (reliability, maintainability, provision for technology upgrade, built-in redundancy, etc.), while providing for human execution of functions (planned and unplanned) critical to servicing missions. With four successful service missions complete, including one initially not planned (the primary mirror repair), the benefits of design-for-sustainment, or life cycle support, throughout all phases of the program becomes quite evident. Without this design approach, it is unlikely that the unanticipated, unplanned mirror repair could even have been attempted, let alone been totally successful.

Risk Management LP 5: For complex programs, the number of stakeholders (government and contractor) demands that the program be structured to cope with high risk factors in many management and technical areas simultaneously. The HST program relied heavily on the contractors (especially Lockheed Missiles and Space Company (LMSC) and Perkin-Elmer (P-E)), each of which "owned" very significant and unique program risk areas. In the critical area of optical systems, NASA depended on LMSC as the overall integrator to manage risk in an area where P-E was clearly the technical expert. Accordingly, NASA relied on LMSC and LMSC relied on P-E with insufficient checks, oversight, and independence of the quality assurance function throughout. While most other risk areas were no doubt managed effectively, lapses here led directly to the HST's going to orbit with the primary mirror defect undetected, in spite of substantial evidence that could have been used to prevent this.

References

Works Cited

Friedman, G.R. and A.P. Sage. 2003. Systems Engineering Concepts: Illustration Through Case Studies. January 19, 2003. Accessed in September 2011. Available at: http://www.afit.edu/cse/docs/Friedman-Sage%20Framework.pdf.

Friedman, G. and A. Sage. 2004. "Case Studies of Systems Engineering and Management in Systems Acquisition." Systems Engineering. 7(1): 84-96.

Mattice, J. "Hubble Space Telescope Case Study." Ft. Belvoir, VA: Defense Acquisition University (DAU). Last modified May 2, 2005. Accessed on November 27, 2012. Available at https://acc.dau.mil/adl/en-US/37600/file/9105/Hubble%20Space%20Telescope%20SE%20Case%20Study%20-%20JJ%20Mattice.pdf.

Primary References

None.

Additional References

None.

< Previous Article | Parent Article | Next Article >

SEBoK v. 1.4 released 29 June 2015

SEBoK Discussion

Please provide your comments and feedback on the SEBoK below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account. Simply type your comment in the text field below and DISQUS will guide you through the login or registration steps. Feedback will be archived and used for future updates to the SEBoK. *If you provided a comment that is no longer listed, that comment has been adjudicated. You can view adjudication for comments submitted prior to SEBoK v. 1.0 at SEBoK Review and Adjudication. Later comments are addressed and changes are summarized in the Letter from the Editor and Acknowledgements and Release History.*

If you would like to provide edits on this article, recommend new content, or make comments on the SEBoK as a whole, please see the SEBoK Sandbox [1].

ENCODED_CONTENT
ODQ1MDAPGRpdiBpZD0iZGlzcXVzX3RocmVhZCI+PC9kaXY+CjxzY3JpcHQgdHlwZT0idGV4dC9qYXZhc2Ny
END_ENCODED_CONTENT

Global Positioning System Case Study

The Global Positioning System (GPS) case study was developed by the United States Air Force Center for Systems Engineering (AF CSE) located at the Air Force Institute of Technology (AFIT). The GPS is a space-based radio-positioning system. A constellation of twenty-four satellites, including three spares, comprise the overall system which provides navigation and timing information to military and civilian users worldwide. GPS satellites, in one of six Earth orbits, circle the globe every twelve hours, emitting continuous navigation signals on two different L-band frequencies. The system consists of two other major segments: a world-wide satellite control network, and the GPS user equipment that can either be carried by a human user or integrated into host platforms such as ships, vehicles, or aircraft.

This case study discussion is based on the original source (O'Brien and Griffin 2007) which provides useful insights into what we might consider a "traditional" SE application. A second Global Positioning System Case Study II looks at the same case study from the perspectives of System of Systems (SoS) (glossary) engineering and Enterprise Systems Engineering (ESE) (glossary).

Domain Background

When looking at the Global Positioning System (GPS), it would be difficult to imagine another system that relie s so heavily upon such a wide range of domains, with the possible exception of the World Wide Web (WWW). Additionally, the various systems operating within these domains must all function together flawlessly to achieve success. It is evident from reading this case study that it directly relates to the following domains:

- aerospace;
- space;
- communications; and
- transportation.

This is also an example of systems of systems (SoS) and is considered an innovative technology.

The GPS case study includes a detailed discussion of the development of the GPS and its components, as well as other applicable areas. The reader of this study will gain an increased understanding of the effect that GPS has on military and commercial industries in the context of the systems engineering support required to achieve success.

Case Study Background

The United States Air Force Center for Systems Engineering (AF CSE), established in 2002 at the Air Force Institute of Technology (AFIT), was tasked to develop case studies focusing on the application of systems engineering principles within various aerospace programs. The GPS case study (O'Brien and Griffin 2007) was developed by AFIT in support of systems engineering graduate school instruction. The cases are structured using the Friedman-Sage framework (Friedman and Sage 2003; Friedman and Sage 2004, 84-96), which decomposes a case into contractor, government, and shared responsibilities in the following nine concept areas:

1. Requirements Definition and Management
2. Systems Architecture Development
3. System/Subsystem Design
4. Verification/Validation
5. Risk Management
6. Systems Integration and Interfaces
7. Life Cycle Support
8. Deployment and Post Deployment
9. System and Program Management

The Friedman-Sage framework (2004) is provided in Appendix A of the case study. This case study is an example where the government - specifically the JPO Systems Engineering Directorate - bore the responsibility for systems integration and configuration management. That is, the government played more than an oversight role in the systems engineering of the GPS system of systems. As mentioned in the case study, JPO developed the CONOPs, mission analysis, requirements and design analysis including security, and developed their own approach to the cryptology methodology. JPO coordinated the Configuration Control Board (CCB) chaired by the Program Director. JPO was also responsible for Level I ICDs and system design configurations; where the contractors were responsible for the system architecture and ICDs within their segment.

Case Study Description

The “Global Positioning System - Systems Engineering Case Study” describes the application of systems engineering during the concept validation, system design and development, and production phases of the GPS program (O'Brien and Griffin 2007). The case examines the applied systems engineering processes, as well as the interactions of the GPS joint program office (JPO), the prime contractors, and the plethora of government agencies that were associated with the program’s development and fielding. The systems engineering process is traced from the initiation of studies and the development of key technologies, which established the vision of a satellite navigation system in the 1960s, through to the multiphase joint-program that resulted in a fully operational capability release in 1995. This case study does not cover system enhancements incorporated through Blocks IIM, IIF, and III.

The GPS case study derived four learning principles (LPs) that explain the more broadly applicable areas of systems engineering knowledge that are addressed by the case study. These four LPs relate strongly to the SEBoK in the following areas:

- enabling individuals (LP1);
- configuration management (LP2);
- enabling the organization (LP3); and
- risk management (LP4).

Additionally, the GPS case study contains a thorough overview of life cycle management and exemplifies systems thinking principles.

Enabling Individuals

Learning Principle 1: Programs must strive to staff key positions with domain experts.

From the program management team, to the systems engineering, design, manufacturing, and operations teams, the individuals on the program were well-versed in their disciplines and all possessed a systems view of the program. While communications, working relationships, and organization were important, it was the ability of the whole team at all levels to understand the implications of their work on the system that was vital. Their knowledge-based approach for decision making had the effect of shortening the decision cycle because the information was understood and the base and alternative solutions were accurately presented.

Configuration Management

Learning Principle 2: The systems integrator must rigorously maintain program baselines.

The joint program office (JPO) retained the role of managing and controlling the system specification and, therefore, the functional baseline. The JPO derived and constructed a mutually agreed to set of system requirements that became the program baseline in 1973. While conducting the development program, the GPS team was able to make performance, risk, cost, and trade analyses against the functional baseline to control both risk and cost. The JPO was fully cognizant of the implications of the functional requirements on the allocated baseline because they managed the interface control working group process. Managing that process gave them first-hand knowledge and insight into the risks at the lowest level. The individual with the system integrator role must rigorously maintain the system specification and functional baseline. There must be appropriate sharing of management and technical responsibilities between the prime contractor and their government counterparts to ensure success.

Enabling the Organization

Learning Principle 3: Achieving consistent and continuous high-level support and advocacy helps funding stability, which impacts systems engineering stability.

Consistent, continuous high-level support provides the requirements and assists funding stability. In this role, the Office of the Secretary of Defense (OSD) provided advocacy and sourced the funding at critical times in the program, promoted coordination among the various services, and reviewed and approved the GPS JPO system requirements. The OSD played the central role in the establishment and survivability of the program. The GPS JPO had clear support from the Director of Defense Development, Research, and Engineering, Dr. Malcolm Currie, and program support from the Deputy Secretary of Defense, Dr. David Packard. Clearly, the armed services – particularly the Navy and the Air Force early on, and later the Army – were the primary users of GPS and the eventual customers. However, each armed service had initial needs for their individual programs, or for the then-current operational navigation systems. Additionally, the secretary of the Air Force provided programmatic support to supply manpower and facilities.

Risk Management

Learning Principle 4: Disciplined and appropriate risk management must be applied throughout the life cycle.

The GPS program was structured to address risk in several different ways throughout the multiphase program. Where key risks were known up front, the contractor and/or the government utilized a classic risk management approach to identify and analyze risk, as well as develop and track mitigation actions. These design (or manufacturing/launch) risks were managed by the office who owned the risks. Identified technical risks were often tracked by technical performance measures (such as satellite weight and software lines of codes) and addressed at weekly chief engineer's meetings.

Serving in the clear role of program integrator allowed the JPO to sponsor risk trade studies at the top level. The JPO would issue study requests for proposals to several bidders for developing concepts and/or preliminary designs. Then, one contractor would be down-selected and the process would continue. This approach provided innovative solutions through competition, as well as helped in defining a lower risk, more clearly defined development program for the fixed-price contracts approach that was being used for development and production.

As the system integrator, the JPO was also closely involved with technical development. To identify unforeseeable unique technical challenges, the JPO would fund studies to determine the optimal approaches to new issues. There were schedule risks associated with the first launch due to unforeseen Block II issues with respect to the space vehicle and control segments (software development). Although a catastrophic event, the Challenger accident actually provided much needed schedule relief. Using decision analysis methodology led the JPO to an alternative approach to develop the expendable launch vehicle for the Block II satellites.

Good communication, facilitated by cooperative working relationships, was a significantly positive (though intangible) factor in the success of the GPS program, regardless of whether it was between the contractors and the government (JPO or other agencies), or between contractors and sub-contractors. A true team environment also played a significant role in reducing risk, especially considering the plethora of government agencies and contractors that were involved in the effort.

Life Cycle Management

The GPS case study takes the reader through the initial concept of GPS (March 1942) all the way to the development, production, and operational capability of the system. The current GPS program traces its heritage to the early 1960s when Air Force Systems Command initiated satellite-based navigation systems analyses conducted by The Aerospace Corporation. The case study follows the execution of the GPS program from the inception of the idea to the full operational capability release on April 27th, 1995. The concentration of the case study is not limited to any particular period, and the learning principles come from various times throughout the program's life.

Systems Thinking

The GPS case study highlights the need for systems thinking throughout. GPS satellites, in one of six Earth orbits, circle the globe every twelve hours. These satellites emit continuous navigation signals on two different L-band frequencies. The system consists of two other major segments: a world-wide satellite control network and the GPS user equipment that can either be carried by a human user, or integrated into host platforms such as ships, vehicles, or aircraft. The ability to conceive, develop, produce, field, and sustain the GPS demands the highest levels of systems thinking.

Summary

The GPS case study is useful for global systems engineering learning and provides a comprehensive perspective on the systems engineering life cycle. The study is applicable for detailed instruction in the following areas:

- enabling individuals;
- configuration management;
- enabling the organization;
- risk management;
- life cycle management; and
- systems thinking.

The GPS case study revealed that key Department of Defense personnel maintained a clear and consistent vision for this unprecedented, space-based navigation capability. The case study also revealed that good fortune was enjoyed by the JPO as somewhat independent, yet critical, space technologies matured in a timely manner.

Although the GPS program required a large degree of integration, both within the system and external to the system amongst a multitude of agencies and contractors, the necessary efforts were taken to achieve success.

Lastly, the reader of the GPS case study will gain an increased understanding of the effect that GPS has on the military and commercial industries in the context of the systems engineering support required to achieve success. The system was originally designed to help "drop five bombs in one hole" which defines the accuracy requirement in context-specific terms. The GPS signals needed to be consistent, repeatable, and accurate to a degree that, when used by munitions guidance systems, would result in the successful delivery of multiple, separately-guided munitions to virtually the identical location anywhere at any time across the planet. Forty to fifty years ago, very few outside of the military recognized the value of the proposed accuracy and most non-military uses of GPS were not recognized before 1990. GPS has increasingly grown in use and is now used every day.

References

Works Cited

Friedman, G.R. and A.P. Sage. 2003. Systems Engineering Concepts: Illustration Through Case Studies. January 19, 2003. Accessed September 2011. Available at: http://www.afit.edu/cse/docs/Friedman-Sage%20Framework.pdf.

Friedman, G. and A. Sage. 2004. "Case Studies of Systems Engineering and Management in Systems Acquisition." *Systems Engineering.* 7(1): p. 84-96.

O'Brien, Patrick J., and John M. Griffin. 4 October 2007. Global Positioning System. Systems Engineering Case Study. Air Force Center for Systems Engineering (AFIT/SY) Air Force Institute of Technology (AFIT). 2950 Hobson Way, Wright-Patterson AFB OH 45433-7765

Primary References

O'Brien, Patrick J., and John M. Griffin. 4 October 2007. Global Positioning System. Systems Engineering Case Study. Air Force Center for Systems Engineering (AFIT/SY) Air Force Institute of Technology (AFIT). 2950 Hobson Way, Wright-Patterson AFB OH 45433-7765

Additional References

none.

< Previous Article | Parent Article | Next Article >

SEBoK v. 1.4 released 29 June 2015

SEBoK Discussion

Please provide your comments and feedback on the SEBoK below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account. Simply type your comment in the text field below and DISQUS will guide you through the login or registration steps. Feedback will be archived and used for future updates to the SEBoK. *If you provided a comment that is no longer listed, that comment has been adjudicated. You can view adjudication for comments submitted prior to SEBoK v. 1.0 at SEBoK Review and Adjudication. Later comments are addressed and changes are summarized in the Letter from the Editor and Acknowledgements and Release History.*

If you would like to provide edits on this article, recommend new content, or make comments on the SEBoK as a whole, please see the SEBoK Sandbox [1].

ENCODED_CONTENT
NTY5OTcPGRpdiBpZD0iZGlzcXVzX3RocmVhZCI+PC9kaXY+CjxzY3JpcHQgdHlwZT0idGV4dC9qYXZhc2Nya
END_ENCODED_CONTENT

Global Positioning System Case Study II

This article highlights some of the differences between the so-called classical, traditional, or conventional systems engineering (SE) approaches and the newer, and, as yet, less defined principles of system of systems (SoS) engineering (SoSE). enterprise systems engineering (ESE), and/or complex systems engineering (CSE) or complex adaptive systems engineering (Gorod et al. 2015). The topic is still somewhat controversial, especially considering those that are sceptical that broader views of SE might work better when one is immersed in trying to cope with our most difficult problems. Indeed, the lack of a unified theory of SE is one of the prime motivations for producing and analysing case studies to develop more knowledge of what seems to work, what does not seem to work, and reasons why, really challenging SE environments.

For addition information, refer to Systems Engineering: Historic and Future Challenges, Systems Engineering and Other Disciplines, Enterprise Systems Engineering, and System of Systems Engineering.

Rather than modifying the previous discussion of the Global Positioning System Case Study in SEBoK, the focus is on comparing and contrasting the older and newer forms of SE by commenting on quotations from the original case study source documents (O'Brien and Griffin 2007).

Preface

The original case study begins by describing systems engineering (SE) principles. For example,

> System requirements are critical to all facets of successful system program development. First, system development must proceed from a well-developed set of requirements. Second, regardless of the evolutionary acquisition approach, the system requirements must flow down to all subsystems and lower-level components. And third, the system requirements must be stable, balanced, and must properly reflect all activities in all intended environments. However, system requirements are not unchangeable. As the system design proceeds, if a requirement or set of requirements is proving excessively expensive to satisfy, the process must rebalance schedule, costs, and performance by changing or modifying the requirements or set of requirements. (O'Brien and Griffin 2007, p. 9)

The Global Positioning System (GPS), including its multi-various applications, was developed over many years as the result of the efforts of a host of contributors. It is very difficult to believe that the classical, traditional or conventional systems engineering approach described in the above paragraph (especially those phrases highlighted in bold by the present authors) was truly responsible for this remarkable achievement that so profoundly impacts our lives. Rather, some more advanced form of systems engineering (SE), that might be called, system of systems (SoS) engineering (SoSE), enterprise systems engineering (ESE), or complex (adaptive) systems engineering (CSE), or a blend and/or combination of these approaches or methodologies, had to be responsible. This premise is supported explicitly and repeatedly in the following case study revision using bold font.

Continuing, the following quoted paragraphs seem flawed in several places highlighted in bold. The bold phrases might be replaced by the phrases in brackets […]. Such brackets might also include other editorial comments of the present authors.

> Systems engineering includes making key system and design trades early in the process to establish the system architecture. These architectural artifacts This architecture can depict any new system, legacy system, modifications thereto, introduction of new technologies, and overall system-level behavior and performance. Modeling and simulation are generally employed to organize and assess architectural system alternatives at this stage. System and subsystem design follows the functional [system] architecture [as defined from a functional point of view]. System architectures designs are modified if elements are too risky, expensive, or time-consuming. (O'Brien and Griffin 2007, p. 9)

A good architecture, once established, should guide systems development, and not change very much, if at all, at least compared to possible changes in the system design, which, of course, can evolve as one learns more about the problem and potential solutions that may increase the system's capability. Thus, it is crucial to not confuse architecture with designs instantiating the architecture, contrary to what seems to be the case in (Ricci, et al. 2013).

> Important to the efficient decomposition and creation of functional and physical architectural designs are the management of interfaces and the integration of subsystems. interface management and integration is applied to subsystems within a system or across a large, complex system of systems. Once a solution is planned, analyzed, designed, and constructed, validation and verification take place to ensure satisfaction of requirements. Definition of test criteria, measures of effectiveness (MOEs), and measures of performance (MOPs) are established as part of the requirements process, taking place well before any component/subsystem assembly design and construction occurs. (O'Brien and Griffin 2007, p. 10)

In the quoted paragraph just above bold phrases note the emphasis on a reductionist approach, reductionism, where great attention is paid to the subsystems and managing the interfaces among them. This is the antithesis of a holistic approach where one concentrates on the whole system, recognizing that it is difficult to identify overall system behavior as depending on any particular subsystem or set of subsystems. In a truly complex system that is continually evolving, the above-mentioned requirements process is flawed because the system is continually changing, i.e., the system is evolutionary; the requirements are either ill-defined at the outset, or are modified because stakeholders change their minds, or become somewhat irrelevant because the system environment changes.

> There are several excellent representations of the [usual traditional or conventional] systems engineering process presented in the literature. These depictions present the current state of the art in maturity and evaluation of the systems engineering process. One can find systems engineering process definitions, guides, and handbooks from the International Council on Systems Engineering (INCOSE), European Industrial Association (EIA), Institute of Electrical and Electronics Engineers (IEEE), and various Department of Defense (DoD) agencies and organizations. They show the process as it should be applied [Really? In all situations?] by today's experienced practitioner. One of these processes, long used by the Defense Acquisition University (DAU), is [a model] not accomplished in a single pass. This iterative and nested process gets repeated to the lowest level of definition of the design and its interfaces. (O'Brien and Griffin 2007, p. 10)

The above description appears to be written with pride without any acknowledgement that this SE methodology might fail to work if applied according to these guidelines, or that there might be new SE techniques that could be more effective in some situations. Again, this reflects a reductionist approach that ignores holism and emergent properties that might not be explained even when thoroughly understanding the systems components and their interactions. On the positive side, the next paragraph suggest how the world is changing and hints that something more is needed. Nevertheless, the advice seems to be oriented toward applying the existing SE discipline more vigorously instead of seeking new methods that might be more effective.

> The DAU model, like all others, has been documented in the last two decades, and has expanded and developed to reflect a changing environment. Systems are becoming increasingly complex internally and more interconnected externally. The process used to develop aircraft and systems of the past was effective at the time. It served the needs of the practitioners and resulted in many successful systems in our inventory. Notwithstanding, the cost and schedule performance of the past programs are replete with examples of well-managed programs and ones with less-stellar execution. As the nation entered the 1980s and 1990s, large DoD and commercial acquisitions experienced overrunning costs and slipping schedules. The aerospace industry and its organizations were becoming larger and were more geographically and culturally distributed. Large aerospace companies have worked diligently to establish common systems engineering practices across their enterprises. However, because of the mega-trend of teaming in large (and some small) programs, these common practices must be understood

> and used beyond the enterprise and to multiple corporations. It is essential that the systems engineering process govern integration, balance, allocation, and verification, and be useful to the entire program team down to the design and interface level. (O'Brien and Griffin 2007, p. 11)

Finally, in the next paragraph there is a suggestion that SE could be made more sophisticated but there is no mention of addressing people problems or advocating a broader transdisciplinary approach.

> Today, many factors overshadow new acquisition; including system-of-systems (SoS) con- text, network centric warfare and operations, and rapid growth in information technology. These factors are driving a more sophisticated systems engineering process with more complex and capable features, along with new tools and procedures. One area of increased focus of the systems engineering process is the informational systems architectural definitions used during system analysis. This process, described in DoD Architectural Framework (DoDAF), emphasizes greater reliance on reusable architectural views describing the system context and concept of operations, interoperability, information and data flows, and network service-oriented characteristics. (O'Brien and Griffin 2007, p. 11)

The last two sections of the systems engineering principles portion of the original case study address case studies themselves, mainly for academic purposes, to help people appreciate systems engineering principles, and the framework used in the case study, namely the rather narrowly defined Friedman-Sage framework that will be discussed briefly in Section II below.

The treatment of the reason for case studies is quite good in that it talks about the benefits of applying systems engineering principles, as highlighted from real-world examples of what works and what does not. Except near the end, where there is allusion to the possibility of new endeavor systems engineering principles, the principles espoused tend to be traditional or conventional.

On the other hand, based upon the original case study (O'Brien and Griffin 2007), if one views the boundary of the GPS system to include primarily the technology associated with the GPS space segment and its controlling ground network, then it can be assumed that system was likely implemented primarily by following traditional or conventional systems engineering processes. If one takes this viewpoint, then all of the above criticism which attempts to point out some of the shortcomings of conventional systems engineering, may seem vacuous at best, or politically incorrect at worst. It may well be that many would rather not denigrate the original GPS case study by exposing it to the possibilities of a broader system engineering approach.

Unless otherwise indicated, as the present authors have already been doing, unchanged quotations from the existing SEBoK are indented below. Modifications to such quotations are shown in brackets [...]; deletions are not necessarily shown explicitly.

Background

> The Global Positioning System (GPS) case study was developed by the United States Air Force Center for Systems Engineering (AF CSE) located at the Air Force Institute of Technology (AFIT). The GPS is a space-based radio-positioning system. A constellation of twenty-four satellites, including three spares, comprise the overall system which provides navigation and timing information to military and civilian users worldwide. GPS satellites, in one of six Earth orbits, circle the globe every twelve hours, emitting continuous navigation signals on two different L-band frequencies. The system consists of two other major segments: a world-wide satellite control network, and the GPS user equipment that can either be carried by a human user or integrated into host platforms such as ships, vehicles, or aircraft.

A user needs to receive signals from at least four GPS satellites simultaneously (satellite orbital positions and terrestrial terrain blockage can be issues that degrade performance) to determine one's position in three dimensions; the altitude determination is typically less accurate than the other two dimensions.

When looking at [GPS], it would be difficult to imagine another system that relies so heavily upon such a wide range of [domains containing systems that must interact effectively to achieve successful GPS operation]. It is evident that [GPS directly relates to many domains and applications including:

- position location and tracking
- time synchronization
- navigation
- transportation
- times of arrival
- air traffic management
- situational awareness
- jam-resistant communications
- business and commerce
- farming
- aerospace
- sensing nuclear detonations from space
- military war-fighting
- targeting
- weapons delivery
- etc.].

[GPS is] an example of [a collaborative (Dahmann, et al. 2008) systems of systems (SoS)]. As such, no one is in charge, and the capabilities (not requirements) flow from the bottom-up, as opposed to top-down.

Purpose

The GPS case study includes a detailed discussion of the development of the GPS and its components, as well as other applicable areas. The reader of this study will gain an increased understanding of the effect that GPS has on military and commercial industries in the context of the systems engineering support required to achieve success.

This may be, but the principal purpose of this revised case study is to suggest a broader view of GPS that discusses signature aspects of SoS, enterprises, and complex systems, and emphasizes SoSE, ESE, and CSE.

[AF CSE] was tasked to develop case studies focusing on the application of [SE] principles within various aerospace programs. The GPS case study [was developed in support of SE] graduate school instruction using the Friedman-Sage framework (Friedman and Sage 2003) (Friedman and Sage 2004).]

However, the Friedman-Sage framework involves only two contractual stakeholders, the Government and the contractor; further, the framework is limited to the traditional or conventional SE life cycle which mainly treats activities in a linear instead of nonlinear fashion; still further, only risks are considered, not a balance of risk and opportunity. Thus, the present authors believe a broader framework embracing SoSE, ESE, and CSE is more appropriate.

Challenges

In the original case study the first highly technical section (Section 2) was the system description. The original idea derived from trying to determine the precise orbital parameters of the first artificial satellites such as Sputnik launched by the Soviets in 1957. Researchers at Johns Hopkins realized the inverse, that if one knew precisely the orbital parameters, the locations of ground stations receiving satellite signals could be determined quite accurately. (O'Brien and Griffin 2007, p. 20)

GPS got its start in the early 70s (O'Brien and Griffin 2007, p. 19) building upon several previous satellite navigation systems. The primary motive was very accurate position information for the purposes of military applications. For example, the U.S. Air Force wanted to deliver nuclear weapons from bombers with unprecedented accuracy and precision. (O'Brien and Griffin 2007, p. 29)

With such an intense interest from the military, the first real challenge, other than the many technical challenges of making GPS work as well as envisioned, might have been the question of how to make GPS available to the civilian community so they could share the benefits. The study claimed that the system was always offered for civilian use, albeit with some charge. After the Korean airliner went astray and got shot down by a Soviet interceptor aircraft, President Reagan made GPS officially available for civilian use free of charge. (O'Brien and Griffin 2007, p. 14)

The second challenge could be associated with preserving precision capabilities for the military only, and relegating course acquisition (C/A) accuracy to the civilian community. (O'Brien and Griffin 2007, p. 15) Later this dichotomy was essentially eliminated with the realization that a differential GPS configuration involving a fixed ground station with a precisely known location will yield great accuracy. (Kee, et al. 1991)

The GPS satellites used space-borne atomic clocks. To alleviate the need for updating these clocks too often a successful effort was initiated to revise the international time standard which ended up using relatively infrequent "leap seconds". (O'Brien and Griffin 2007, p. 23) Even these are still annoying for many other applications, such as the continual need to achieve precise synchronization of frequency hopping radios.

An organizational challenge of inter-service rivalries was overcome with the formation of the Joint Program Office (JPO). (O'Brien and Griffin 2007, p. 25)

In the early days of satellite communication systems, for example, the satellites were quite small and low powered while the terminals were large and high-powered. By the time GPS came along, the satellites are getting bigger and more sophisticated. Then the challenge to develop relatively low-cost terminals, particularly for mobile users, greatly increased. (O'Brien and Griffin 2007, p. 29)

A small but interesting challenge was the definition of system of systems (SoS). It was decided that GPS was an SoS because it involved three independent systems, namely, the space vehicle (SV), the control segment (CS), and the user equipment (UE), that "merely" had to interface with each other. (O'Brien and Griffin 2007, p. 30)

Continually changing requirements is usually a problem, although in this case the requirements did not change as often as they could have. (O'Brien and Griffin 2007, p. 31)

Difficulties of defining and updating the many GPS interfaces was largely overcome by the GPS program director, Col. Brad Parkinson, when he convinced his own management, Gen. Schultz at Space and Missile Systems Office (SAMSO) (which eventually became the Space Division) that GPS ought to be defined solely by the signal-structure-in-space and not the physical interfaces. (O'Brien and Griffin 2007, p. 31)

Systems Engineering Practices

> Although the systems engineering process in Phase I has been discussed previously, this section will expand on the concepts. For example, one of the user equipment contractors was technically competent, but lacked effective management. The JPO strongly suggested that a systems engineering firm be hired to assist the contractor in managing program and they agreed. (O'Brien and Griffin 2007, p. 42)

There did not seem to be any mention of what SE firm was hired, if any. The Aerospace Corporation, a non-profit Federally Funded Research and Development Center (FFRDC), which had such a key role in the run-up to GPS was also prominently and centrally involved in development phase of this humungous project. (O'Brien and Griffin 2007, pp. 20, 22, 25, 33, 34, 40, 41, 44, 48, 50-52, 56, 57, 62, 63, 64, 66, 67, 71)

Lessons Learned

> Communications was a key ingredient that was fostered throughout GPS development. (O'Brien and Griffin 2007, p. 71)

Yes, from reading the original case study there seems to have been a lot of cooperation among the various organizations, more so than might have been expected in a less compelling case.

> Several precepts or foundations of the Global Positioning Satellite program are the reasons for its success. These foundations are instructional for today's programs because they are thought-provoking to those who always seek insight into the program's progress under scrutiny. These foundations of past programs are, of course, not a complete set of necessary and sufficient conditions. For the practitioner, the successful application of different systems engineering processes is required throughout the continuum of a program, from the concept idea to the usage and eventual disposal of the system. Experienced people applying sound systems engineering principles, practices, processes, and tools are necessary every step of the way. Mr. Conley, formerly of the GPS JPO, provided these words: "Systems engineering is hard work. It requires knowledgeable people who have a vision of the program combined with an eye for detail." (O'Brien and Griffin 2007, p. 72)

In very complex systems engineering efforts of this type, it is also important to explore new techniques that attempt to deal with "soft" issues involving people. Those that seem to work can be added to the systems engineering process collection.

> Systems engineering played a major role in the success of this program. The challenges of integrating new technologies, identifying system requirements, incorporating a system of systems approach, interfacing with a plethora of government and industry agencies, and dealing with the lack of an operational user early in the program formation required a strong, efficient systems engineering process. The GPS program embedded systems engineering in their knowledge-base, vision, and day-to-day practice to ensure proper identification of system requirements. It also ensured the allocation of those requirements to the almost-autonomous segment developments and beyond to the subcontractor/vendor level, the assessments of new requirements, innovative test methods to verify design performance to the requirements, a solid concept of operations/mission analysis, a cost-benefit analysis to defend the need for the program, and a strong system integration process to identify and control the "hydra" of interfaces that the program encountered. The program was able to avoid major risks by their acquisition strategy, the use of trade studies, early testing of concept designs, a detailed knowledge of the subject matter, and the vision of the program on both the government and contractor side. (O'Brien and Griffin 2007, p. 72)

This well summarizes the successful systems engineering approach utilized in GPS. Another element of achieving overall balance is the pursuit of opportunities as the "flipside" of risk mitigation.

Finally, here is the list of academic questions offered in original case study.

> QUESTIONS FOR THE STUDENT (O'Brien and Griffin 2007, p. 73) The following questions are meant to challenge the reader and prepare for a case discussion.

- Is this program start typical of an ARPA/ DARPA funded effort? Why or why not?
- Have you experiences similar or wildly different aspects of a Joint Program?
- What were some characteristics that should be modeled from the JPO?
- Think about the staffing for the GPS JPO. How can this be described? Should it be duplicated in today's programs? Can it?
- Was there anything extraordinary about the support for this program?
- What risks were present throughout the GPS program. How were these handled?
- Requirement management and stability is often cited as a central problem in DoD acquisition. How was this program like, or [un]like, most others?

- Could the commercial aspects of the User Equipment be predicted or planned? Should the COTS aspect be a strategy in other DoD programs, where appropriate? Why or why not?

  Other questions might be: What possible influences did the demand for or offering to the public of this GPS capability entail?What differences in the development of GPS might have emerged if the public was more aware of the potential applications for their benefit at the outset?

References

Works Cited

Dahmann, J. S., George Rebovich, Jr., and Jo Ann Lane. November 2008. “Systems Engineering for Capabilities.” CrossTalk, The Journal of Defense Software Engineering. http://www.stsc.hill.af.mil/crosstalk/2008/11/index.html.Accessed 12 May 2015).

Friedman, G.R., and A.P. Sage. 19 January 2003. Systems Engineering Concepts: Illustration Through Case Studies. Accessed September 2011. http://www.afit.edu/cse/docs/Friedman-Sage%20Framework.pdf.

Gorod, A., B. E. White, V. Ireland, S. J. Gandhi, and B. J. Sauser. 2015. Case Studies in System of Systems, Enterprise Systems, and Complex Systems Engineering. Boca Raton, FL: CRC Press, Taylor & Francis Group. 2015. http://www.taylorandfrancis.com/books/details/9781466502390/.Accessed 8 May 2015

Friedman, G.R., and A.P. Sage. 2004. “Case Studies of Systems Engineering and Management in Systems Acquisition.” Systems Engineering 7(1): 84-96.

Kee, Changdon, Bradford W. Parkinson, Penina Axelrad. Summer 1991. “Wide Area Differential GPS.” Journal of The Institute of Navigation 38(2): 123-46.

O’Brien, Patrick J., and John M. Griffin. 4 October 2007. Global Positioning System. Systems Engineering Case Study. Air Force Center for Systems Engineering (AFIT/SY) Air Force Institute of Technology (AFIT). 2950 Hobson Way, Wright-Patterson AFB OH 45433-7765.

Ricci, Nicola, Adam M. Ross, Donna H. Rhodes, and Matthew E. Fitzgerald. 2013. “Considering Alternative Strategies for Value Sustainment in Systems-of-Systems.” 6th IEEE International Systems Conference (SysCon). 15-18 April. Orlando, FL.

Primary References

Gorod, A., B. E. White, V. Ireland, S. J. Gandhi, and B. J. Sauser. 2015. Case Studies in System of Systems, Enterprise Systems, and Complex Systems Engineering. Boca Raton, FL: CRC Press, Taylor & Francis Group. 2015. http://www.taylorandfrancis.com/books/details/9781466502390/.Accessed 12 May 2015

Additional References

None

< Previous Article | Parent Article | Next Article>

SEBoK v. 1.4 released 29 June 2015

SEBoK Discussion

Please provide your comments and feedback on the SEBoK below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account. Simply type your comment in the text field below and DISQUS will guide you through the login or registration steps. Feedback will be archived and used for future updates to the SEBoK. *If you provided a comment that is no longer listed, that comment has been adjudicated. You can view adjudication for comments submitted prior to SEBoK v. 1.0 at SEBoK Review and Adjudication. Later comments are addressed and changes are summarized in the Letter from the Editor and Acknowledgements and Release History.*

If you would like to provide edits on this article, recommend new content, or make comments on the SEBoK as a whole, please see the SEBoK Sandbox [1].

ENCODED_CONTENT
MTQwMDgPGRpdiBpZD0iZGlzcXVzX3RocmVhZCI+PC9kaXY+CjxzY3JpcHQgdHlwZT0idGV4dC9qYXZhc2Ny
END_ENCODED_CONTENT

Medical Radiation Case Study

This case study presents system and software engineering issues relevant to the accidents associated with the Therac-25 medical linear accelerator that occurred between 1985 and 1988. The six accidents caused five deaths and serious injury to several patients. The accidents were system accidents that resulted from complex interactions between hardware components, controlling software, and operator functions.

Domain Background

Medical linear accelerators, devices used to treat cancer, accelerate electrons to create high energy beams that can destroy tumors. Shallow tissue is treated with the accelerated electrons. The electron beam is converted to X-ray photons to reach deeper tissues. Accidents occur when a patient is delivered an unsafe amount of radiation.

A radiation therapy machine is controlled by software that monitors the machine's status, accepts operator input about the radiation treatment to be performed, and initializes the machine to perform the treatment. The software turns the electron beam on in response to an operator command. The software turns the beam off whenever the treatment is complete, the operator requests the beam to shutdown, or when the hardware detects a machine malfunction. A radiation therapy machine is a reactive system in which the system's behavior is state dependent and the system's safety depends upon preventing entry into unsafe states. For example, the software controls the equipment that positions the patient and the beam. The positioning operations can take a minute or more to execute, thus it is unsafe to activate the electron beam while a positioning operation is in process.

In the early 1980s, Atomic Energy of Canada (AECL) developed the Therac-25, a dual-mode (X-rays or electrons) linear accelerator that can deliver photons at 25 megaelectron volts (MeV) or electrons at various energy levels. The Therac-25 superseded the Therac-20, the previous 20-MeV dual mode accelerator with a history of successful clinical use. The Therac-20 used a DEC PDP-11 (Digital Equipment Corporation Programmed Data Processor) minicomputer for computer control and featured protective circuits for monitoring the electron beam, as well as mechanical interlocks for policing the machine to ensure safe operation. AECL decided to increase the responsibilities of the Therac-25 software for maintaining safety and eliminated most of the hardware safety mechanisms and interlocks. The software, written in PDP-11 assembly language, was partially reused from earlier products in the Therac product line. Eleven Therac-25s were installed at the time of the first radiation accident in June 1985.

The use of radiation therapy machines has increased rapidly in the last 25 years. The number of medical radiation machines in the United States in 1985 was approximately 1000. By 2009 the number had increased to approximately 4450. Some of the types of system problems found in the Therac-25 may be present in the medical radiation devices currently in use. References to more recent accidents are included below.

Case Study Background

The Therac-25 accidents and their causes are well documented in materials from the U.S. and Canadian regulatory agencies (e.g., the U.S. Food and Drug Administration (FDA) and the Canadian Bureau of Radiation and Medical Devices) and in depositions associated with lawsuits brought against AECL. An article by Leveson and Turner (1993) provides the most comprehensive, publicly available description of the accident investigations, the causes of the accidents, and the lessons learned relevant to developing systems where computers control dangerous devices.

Case Study Description

The Therac-25 accidents are associated with the non-use or misuse of numerous system engineering practices, especially system verification and validation, risk management, and assessment and control. In addition, numerous software engineering good practices were not followed, including design reviews, adequate documentation, and comprehensive software unit and integration tests.

The possibility of radiation accidents increased when AECL made the systems engineering decision to increase the responsibilities of the Therac-25 software for maintaining safety and eliminated most of the hardware safety mechanisms and interlocks. In retrospect, the software was not worthy of such trust. In 1983 AECL performed a safety assessment on the Therac-25. The resulting fault tree did include computer failures, but only those associated with hardware; software failures were not considered in the analysis.

The software was developed by a single individual using PDP-11 assembly language. Little software documentation was produced during development. An AECL response to the FDA indicated the lack of software specifications and of a software test plan. Integrated system testing was employed almost exclusively. Leveson and Turner (1993) described the functions and design of the software and concluded that there were design errors in how concurrent processing was handled. Race conditions resulting from the implementation of multitasking also contributed to the accidents.

AECL technical management did not believe that there were any conditions under which the Therac-25 could cause radiation overdoses, and this belief was evident in the company's initial responses to accident reports. The first radiation overdose accident occurred in June 1985 at the Kennestone Regional Oncology Center in Marietta, Georgia, where the Therac-25 had been operating for about 6 months. The patient who suffered the radiation overdose filed suit against the hospital and AECL in October 1985. No AECL investigation of the incident occurred and FDA investigators later found that AECL had no mechanism in place to follow up potential reports of suspected accidents. Additionally, other Therac-25 users received no information that an accident had occurred.

Two more accidents occurred in 1985, including a radiation overdose at Yakima Valley Memorial Hospital in Yakima, Washington that resulted in an accident report to AECL. The AECL technical support supervisor responded to the hospital in early 1986: "After careful consideration, we are of the opinion that this damage could not have been produced by any malfunction of the Therac-25 or by any operator error… there have apparently been no other instances of similar damage to this or other patients."

In early 1986 there were two accidents at the East Texas Cancer Center in Tyler, Texas, both of which resulted in the death of the patient within a few months. On March 21, 1986 the first massive radiation overdose occurred, though the extent of the overdose was not realized at the time. The Therac-25 was shut down for testing the day after the accident. Two AECL engineers, one from the plant in Canada, spent a day running machine tests but could not reproduce the malfunction code observed by the operator at the time of the accident. The home office engineer explained that it was not possible for the Therac-25 to overdose a patient. The hospital physicist, who supervised the use of the machine, asked AECL if there were any other reports of radiation overexposure. The AECL quality assurance manager told him that AECL knew of no accidents involving the Therac-25.

On April 11, 1986 the same technician received the same malfunction code when an overdose occurred. Three weeks later the patient died; an autopsy showed acute high-dose radiation injury to the right temporal lobe of the brain and to the brain stem. The hospital physicist was able to reproduce the steps the operator had performed and measured the high radiation dosage delivered. He determined that data-entry speed during editing of the treatment script was the key factor in producing the malfunction code and the overdose. Examination of the portion of the code responsible for the Tyler accidents showed major software design flaws. Levinson and Turner (1993) describe in detail how the race condition occurred in the absence of the hardware interlocks and caused the overdose. The first report of the Tyler accidents came to the FDA from the Texas Health Department. Shortly thereafter, AECL provided a medical device accident report to the FDA discussing the radiation overdoses in Tyler.

On May 2, 1986 the FDA declared the Therac-25 defective and required the notification of all customers. AECL was required to submit to the FDA a corrective action plan for correcting the causes of the radiation overdoses. After multiple iterations of a plan to satisfy the FDA, the final corrective action plan was accepted by the FDA in the summer of 1987. The action plan resulted in the distribution of software updates and hardware upgrades that reinstated most of the hardware interlocks that were part of the Therac-20 design.

AECL settled the Therac-25 lawsuits filed by patients that were injured and by the families of patients who died from the radiation overdoses. The total compensation has been estimated to be over $150 million.

Summary

Leveson and Turner (1993) describe the contributing factors to Therac-25 accidents: “We must approach the problems of accidents in complex systems from a systems-engineering point of view and consider all contributing factors." For the Therac-25 accidents, the contributing factors included

- management inadequacies and a lack of procedures for following through on all reported incidents;
- overconfidence in the software and the resulting removal of hardware interlocks (causing the software to be a single point of failure that could lead to an accident);
- less than acceptable software engineering practices; and
- unrealistic risk assessments along with over confidence in the results of those assessments.

Recent Medical Radiation Experience

Between 2009 and 2011, The New York Times published a series of articles by Walter Bogdanich on the use of medial radiation, entitled “Radiation Boom" (2011).

The following quotations are excerpts from that series:

> *Increasingly complex, computer-controlled devices are fundamentally changing medical radiation, delivering higher doses in less time with greater precision than ever before.” But patients often know little about the harm that can result when safety rules are violated and ever more powerful and technologically complex machines go awry. To better understand those risks, The New York Times examined thousands of pages of public and private records and interviewed physicians, medical physicists, researchers and government regulators. The Times found that while this new technology allows doctors to more accurately attack tumors and reduce certain mistakes, its complexity has created new avenues for error — through software flaws, faulty programming, poor safety procedures or inadequate staffing and training. . . .*
>
> *Linear accelerators and treatment planning are enormously more complex than 20 years ago,’ said Dr. Howard I. Amols, chief of clinical physics at Memorial Sloan-Kettering Cancer Center in New York. But*

> *hospitals, he said, are often too trusting of the new computer systems and software, relying on them as if they had been tested over time, when in fact they have not. . . .*
>
> *Hospitals complain that manufacturers sometimes release new equipment with software that is poorly designed, contains glitches or lacks fail-safe features, records show. Northwest Medical Physics Equipment in Everett, Wash., had to release seven software patches to fix its image-guided radiation treatments, according to a December 2007 warning letter from the F.D.A. Hospitals reported that the company's flawed software caused several cancer patients to receive incorrect treatment, government records show.*

References

Works Cited

Bogdanich, W. 2011. Articles in the "Radiation Boom" series. *New York Times.* June 2009-February 2011. Accessed November 28, 2012. Available: http://topics.nytimes.com/top/news/us/series/radiation_boom.

Leveson, N.G., and C.S. Turner. 1993. "An Investigation of the Therac-25 Accidents." *Computer*. 26 (7): 18-41.

Primary References

None.

Additional References

None.

< Previous Article | Parent Article | Next Article >

SEBoK v. 1.4 released 29 June 2015

SEBoK Discussion

Please provide your comments and feedback on the SEBoK below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account. Simply type your comment in the text field below and DISQUS will guide you through the login or registration steps. Feedback will be archived and used for future updates to the SEBoK. *If you provided a comment that is no longer listed, that comment has been adjudicated. You can view adjudication for comments submitted prior to SEBoK v. 1.0 at SEBoK Review and Adjudication. Later comments are addressed and changes are summarized in the Letter from the Editor and Acknowledgements and Release History.*

If you would like to provide edits on this article, recommend new content, or make comments on the SEBoK as a whole, please see the SEBoK Sandbox [1].

ENCODED_CONTENT
MTU2NjQPGRpdiBpZD0iZGlzcXVzX3RocmVhZCI+PC9kaXY+CjxzY3JpcHQgdHlwZT0idGV4dC9qYXZhc2Nya
END_ENCODED_CONTENT

FBI Virtual Case File System Case Study

This case study presents systems and software engineering issues encountered in the Federal Bureau of Investigation (FBI) Virtual Case File (VCF) project in the period between 2000-2005. VCF development was abandoned in 2005 after over $170 million had been spent.

Domain Background

The FBI is an organization within the United States Department of Justice (DoJ) consisting of 23 divisions, including counterintelligence, criminal investigation, and cyber crime. The Bureau's 12,400 agents investigate everything from counter-terrorism leads to kidnappings. They interview witnesses, develop informants, conduct surveillance, hunt for clues, and collaborate with local law enforcement to find and arrest criminals. Agents document every step and methodically build case files. They spend a tremendous amount of time processing paperwork. This system of forms and approvals stretches back to the 1920s when forms for all of the bureau's investigative reports were standardized.

In 2000, the Bureau had hundreds of standardized paper forms and obsolete information technology (IT) systems. The FBI’s 13,000 computers could not run modern software. Most of the agency offices were connected to the FBI Intranet with links operating at about the speed of a 56 kilobits-per-second modem. Agents could not e-mail U.S. Attorneys, federal agencies, local law enforcement, or each other; instead, they typically sent case-related information by fax. The agency’s problems in 2000 were summarized in the *9/11 Commission Report*: "the FBI's information systems were woefully inadequate. The FBI lacked the ability to know what it knew; there was no effective mechanism for capturing or sharing its institutional knowledge" (National Commission on Terrorist Acts upon the United States 2004).

In September 2000, Congress approved $380 million over three years for what was then called the FBI Information Technology Upgrade Program. Eventually divided into three parts, the program became known as the Trilogy Information Technology Modernization Program. The first part would provide all 56 FBI field offices with updated computer terminals, as well as new hardware such as scanners, printers, and servers. The second part would re-implement the FBI Intranet to provide secure local area and wide area networks, allowing agents to share information with their supervisors and each other. The third part was intended to replace the FBI's investigative software applications, including the obsolete Automated Case Support (ACS) system.

In June 2001, the FBI awarded a contract to develop the investigative software applications of Trilogy to Science Applications International Corporation (SAIC) over a three year period. The purpose of the software to be developed was to

- provide the capability to find information in FBI databases without having prior knowledge of its location, and to search all FBI databases with a single query through the use of search engines;
- Web-enable the existing investigative applications;
- improve capabilities to share information inside and outside the FBI;
- provide access to authorized information from both internal and external databases; and
- allow the evaluation of cases and crime patterns through the use of commercial and FBI-enhanced analytical and case management tools.

After the September 11 terrorist attacks, the inability of FBI agents to share the most basic information about al Qaeda's U.S. activities was front-page news. Within days, the FBI's obsolete technology infrastructure was being discussed in Congress and the FBI was under intense pressure to improve its information sharing capabilities. On September 4, 2001, Robert S. Mueller III became FBI director, and, in the face of intense public and congressional pressure, Mueller accelerated the Trilogy program. The planned three year period to develop the investigative software was considered politically unacceptable. In January 2002, the FBI requested an additional $70 million to accelerate Trilogy; Congress went further, approving $78 million.

Providing web-enablement of the existing but antiquated and limited ACS system would not provide the investigative case management capabilities required to meet the FBI's post-September 11 mission. In December 2001, the FBI asked SAIC to stop building a Web-based front end for the old programs. Instead, SAIC was asked to devise a new case management system, the Virtual Case File (VCF), to replace ACS. The VCF would contain a major new application, database, and graphical user interface. In order to make both criminal and terrorist investigation information readily accessible throughout the FBI, major changes to the standardized FBI processes would be required. This case study focuses on the VCF component of the Trilogy program.

Case Study Background

The most complete description of the development of the VCF is the report by the DoJ Office of the Inspector General (OIG). The OIG reports to the Attorney General and is independent of the FBI organizations responsible for the Trilogy program. The introduction to the report states, "We conducted this audit to assess the FBI's progress in meeting cost, schedule, technical, and performance targets for the three components of Trilogy. We also examined the extent to which Trilogy will meet the FBI's current and longer-term IT needs" (OIG 2004).

An IEEE *Spectrum* article complements the OIG audit report by providing detailing the development of the VCF requirements, the contractor's activities, and the project management failures by both the FBI and the contractor. The contractor's viewpoint is presented in testimony given before a subcommittee of the U.S. Senate Appropriations Committee.

These materials, in total, provide a comprehensive view of the VCF program and the reasons for its failure.

Case Study Description

In the political environment following the 9/11 attacks, funding for the VCF project was never a problem. By early 2002, SAIC and the FBI committed to creating an entirely new case management system in 22 months. High-level funding enabled the project to continue gaining momentum in spite of the problems it encountered. The scheduling for the VCF project focused on what was desired, not what was possible. Trilogy's scope grew by approximately 80% from the initial project baseline (Moore 2010).

The reasons for the failure of the VCF project are associated with the non-use or misuse of numerous system engineering practices, especially within stakeholder requirements, system requirements, planning, assessment and control, and risk management. Given the political pressures following the 9/11 attacks, the schedule was accelerated to the point that it was nearly impossible for the developers to follow an appropriate systems engineering process.

The FBI cycled through five people in the role of Chief Information Officer in four years and most decisions were made by committees. In order to compress the schedule, the FBI even proposed replacing the ACS with the VCF over a weekend using an IT procedure called a "flash cut-over." In this proposed implementation, the ACS system would be taken offline and entirely replaced by VCF. Once the cut-over happened, there would be no mechanism to return to ACS, even if the VCF did not work properly.

SAIC worked under a cost-plus-award-fee contract for the VCF as the scope of the project was undefined in early 2002 when work began. Given the schedule pressures, the FBI believed that there was no time to develop formal requirements (glossary), validate them with the various FBI user communities, and then estimate the cost and time required to develop the VCF. The SAIC contract did not require specific completion milestones and the cost-plus contract allowed the scope to increase. VCF was a case of not getting the requirements sufficiently defined in terms of completeness and correctness. The continuous redefinition of requirements had a cascading effect on what had already been designed and produced. Once there was demonstrable software, change requests started arriving—roughly 400 from December 2002 to December 2003.

The new FBI Intranet was specified during 2001, before the start of the VCF project and with little understanding of the network traffic that would result from information sharing. By early 2003, the FBI began to realize how taxing the network traffic would be once all 22,000 users came online. The requirements for the FBI Intranet were modified based on the best guesses for the bandwidth that would be required when the VCF was fully operational. By early 2004, the new FBI Intranet was in operation, although the VCF software was far from complete.

In reaction to the time pressure, SAIC broke its VCF development group into eight teams working in parallel on different functional elements of the program. However, this posed many integration challenges and the eight threads would later prove too difficult for SAIC to combine into a single system. By the time VCF was canceled, SAIC had developed over 700,000 lines of software based upon an incomplete set of requirements that were documented in an 800-page volume.

Summary

The OIG summarizes its conclusions as

> *Various reasons account for the delays and associated cost increases in the Trilogy project, including:*

- *poorly defined and slowly evolving design requirements,*
- *contracting weaknesses,*
- *IT investment management weaknesses,*
- *lack of an Enterprise Architecture,*
- *lack of management continuity and oversight,*
- *unrealistic scheduling of tasks,*
- *lack of adequate project integration, and*
- *inadequate resolution of issues raised in our previous reports on Trilogy. . . .*

> *According to the Government Accountability Office (GAO), an Enterprise Architecture is a set of descriptive models such as diagrams and tables that define, in business and technology terms, how an organization operates today, how it intends to operate in the future, and how it intends to invest in technology to transition from today's operational environment to tomorrow's. . . .*
>
> *As of early 2005 the FBI's operations remain significantly hampered due to the poor functionality and lack of information-sharing capabilities of its current IT systems. . . .* (OIG 2005)
>
> In May 2005, FBI director Mueller announced Sentinel, a four-phase, four-year project intended to fulfill the purpose of VCF and provide the Bureau with a web-based case and records management system. During the previous five years, commercial case management software had become available; as a result, Sentinel is intended to utilize commercial off-the-shelf (COTS) software. A report by the OIG in late 2009 describes Sentinel and its status at that time. Sentinel was put on line for all employees on July 1, 2012, and it ended up at $451 million and 2 1/2 years overdue (Yost 2012).

References

Works Cited

Moore, S. 2010. "The Failure of the FBI's Virtual Case File Project." Strategic PPM: Project and Portfolio Management, last modified April 5, accessed on September 11, 2011. Available at http://strategicppm.wordpress.com/2010/04/05/the-fbis-virtual-case-file-project-and-project-failure.

National Commission on Terrorist Attacks upon the United States. 2004. *The 9/11 Commission Report: Final Report of the National Commission on Terrorist Attacks Upon the United States*. New York, NY, USA: W. W. Norton & Company.

Office of the Inspector General. 2005. *The Federal Bureau of Investigation's Management of the Trilogy Information Technology Project.* Washington, DC, USA: United States Department of Justice. Audit Report 05-07. February 2005. Accessed on September 11, 2011. Available at http://www.justice.gov/oig/reports/FBI/a0507.

Office of the Inspector General. 2009. *Sentinel Audit V: Status of the Federal Bureau of Investigation's Case Management System*. Washington, DC, USA: U.S. Department of Justice. Audit Report 10-03. November 2009. Accessed on September 11, 2011. Available at http://www.justice.gov/oig/reports/FBI/a1003_redacted.pdf.

Yost, P. 2012. "'Sentinel', New FBI Computer System, Finally Tracking Cases -- Years Late and Millions Over Budget." Washington, DC, USA. Accessed on August 6, 2012. Available at http://www.huffingtonpost.com/2012/07/31/sentinel-fbi_n_1725958.html.

Primary References

None.

Additional References

Goldstein, H. 2005. "Who Killed the Virtual Case File?" IEEE *Spectrum*. September. Accessed at September 11, 2011. Available at http://spectrum.ieee.org/computing/software/who-killed-the-virtual-case-file.

Testimony before the Subcommittee on Commerce, Justice, State and the Judiciary, U.S. Senate Committee on Appropriations, February 3, 2005 (statement of Arnold Punaro, Executive Vice President, Science Applications International Corporation).

< Previous Article | Parent Article | Next Article >

SEBoK v. 1.4 released 29 June 2015

SEBoK Discussion

Please provide your comments and feedback on the SEBoK below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account. Simply type your comment in the text field below and DISQUS will guide you through the login or registration steps. Feedback will be archived and used for future updates to the SEBoK. *If you provided a comment that is no longer listed, that comment has been adjudicated. You can view adjudication for comments submitted prior to SEBoK v. 1.0 at SEBoK Review and Adjudication. Later comments are addressed and changes are summarized in the Letter from the Editor and Acknowledgements and Release History.*

If you would like to provide edits on this article, recommend new content, or make comments on the SEBoK as a whole, please see the SEBoK Sandbox [1].

ENCODED_CONTENT
MTg3NTgPGRpdiBpZD0iZGlzcXVzX3RocmVhZCI+PC9kaXY+CjxzY3JpcHQgdHlwZT0idGV4dC9qYXZhc2NyaX
END_ENCODED_CONTENT

MSTI Case Study

The Miniature Seeker Technology Integration (MSTI) spacecraft was the first of its kind: a rapid development spacecraft, designed and launched in one year. As an aerospace example for a satellite application, the case study, "M.S.T.I.: Optimizing the Whole System" (Grenville, Kleiner, and Newcomb 2004), describes the project's systems engineering approach. Driven by an aggressive schedule, MSTI optimized over the whole project, rather than allowing sub-optimizations at the component level. As a partnership with Phillips Laboratories, the Jet Propulsion Laboratory (JPL), and Spectrum Astro, MSTI went into orbit on November 21, 1992. The MSTI-1 succeeded in meeting all primary mission objectives, surpassing the 6-day data collection mission requirement.

Domain Background

There are many case study examples for aerospace systems. This case is of particular interest because it highlights mechanisms which enabled successful performance following an aggressive schedule. Since this rapid development spacecraft was designed and launched in one year, new ways of structuring the project were necessary. Within this domain, the MSTI project used an innovative approach. Practices from this project led to the Mission Design Center and the System Test Bed at JPL.

Case Study Background

This case study was developed in support of the National Aeronautics and Space Administration (NASA) Program and Project Management Initiative by authors at the Virginia Polytechnic Institute and State University and Scientific Management, Inc. The case study was developed in the interest of continuously improving program and project management at NASA (NASA 2010). Research for this case included comprehensive literature review and detailed interviews. The project was selected based on the potential for providing lessons learned.

Case Study Description

The MSTI case study illustrates many principles described in the Systems Engineering Body of Knowledge (SEBoK). The MSTI team had to make adjustments to the traditional approach to spacecraft development in order to stay within budget and to meet the aggressive timeline of bringing a spacecraft from conception to launch within one year. The team realized that they were "building Porsches not Formula 1s"(Grenville, Kleiner, Newcomb 2004). Meeting the schedule was a crucial factor that affected all decisions. The SEBoK knowledge area on life cycle models describes life cycle design in more detail.

The team took advantage of existing hardware architectures for their architectural design to expedite the project. In addition, at each design phase, the whole system was optimized instead of optimizing subsystems, and the level of optimization at the subsystem level was reduced. A hardware-in-the-loop test bed was used throughout the project, which expedited system integration.

The schedule was maintained only at a high level in the project management office, and the costs were managed using a cost reporting technique for "cost to completion." Rather than report on past spending, the Responsible Engineering Authorities (REAs) were expected to continually evaluate their ability to complete their tasks within projected costs. Faster procurement was achieved using the Hardware Acquisition Team, where a technical team member was matched with a procurement representative for each design function. This pair wrote the specifications together and initiated the purchase requisitions.

From the organizational perspective, increased responsibility and accountability were given to each team member. Individuals took ownership of their work and the decision process was streamlined. The team made more "good decisions," rather than optimal decisions. The team was collocated, and daily meetings were used to assign daily tasks and keep the team focused on the launch. The standard Problem Failure Report (PFR) was streamlined and electronic reports provided snapshots of the resolved and outstanding PFRs. The report helped REAs stay on top of potential problem areas. REAs were responsible for looking forward on the project horizon and notifying the team of any potential problem areas.

The first satellite in the MSTI series, MSTI-1, was launched on November 21, 1992. The spacecraft weighed 150 kg and was built for $19M in less than 12 months. Over 200,000 photographs were returned from the spacecraft. From a project management standpoint, all mission objectives were completed.

In addition, MSTI had a lasting legacy. Faster procurement developed into an approach JPL now calls "Fast Track Procurement." Hardware acquisition teams are used often in JPL projects. The hardware-in-the-loop test bed was the precursor to the Flight System Test Bed at JPL. Team members moved up quickly in JPL due to the increased responsibility and authority they were given on the MSTI project.

Summary

MSTI demonstrated that an aggressive schedule can be used to design low earth-orbiting spacecraft to optimize the full system. The MSTI experience changed JPL's culture and their approach to spacecraft development and mission management. The insights from this case study example can help both students and practitioners better understand principles described in the SEBoK.

References

Works Cited

Grenville, D., B.M. Kleiner, and J.F. Newcomb. 2004. *M.S.T.I., Optimizing the Whole System.* Blacksburg, VA: Virginia Polytechnic Institute, case study developed in support of the NASA Program and Project Management Initiative. 1-27. Accessed June 3, 2011. Available at http:/ / www. nasa. gov/ pdf/ 293212main_58529main_msti_casestudy_042604.pdf.

NASA. 2010. *A Catalog of NASA-Related Case Studies,* version 1.6. Compiled by the Office of the Chief Knowledge Officer, Goddard Space Flight Center, MD, USA: National Aeronautics and Space Administration (NASA). Accessed June 3, 2011. Available at http:/ / www. nasa. gov/ centers/ goddard/ pdf/ 450420main_NASA_Case_Study_Catalog.pdf.

Primary References

None.

Additional References

None.

< Previous Article | Parent Article | Next Article >

SEBoK v. 1.4 released 29 June 2015

SEBoK Discussion

Please provide your comments and feedback on the SEBoK below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account. Simply type your comment in the text field below and DISQUS will guide you through the login or registration steps. Feedback will be archived and used for future updates to the SEBoK. *If you provided a comment that is no longer listed, that comment has been adjudicated. You can view adjudication for comments submitted prior to SEBoK v. 1.0 at SEBoK Review* *and Adjudication. Later comments are addressed and changes are summarized in the Letter from the Editor and Acknowledgements and Release History.*

If you would like to provide edits on this article, recommend new content, or make comments on the SEBoK as a whole, please see the SEBoK Sandbox [1].

ENCODED_CONTENT
MjAwNTYPGRpdiBpZD0iZGlzcXVzX3RocmVhZCI+PC9kaXY+CjxzY3JpcHQgdHlwZT0idGV4dC9qYXZhc2Nya
END_ENCODED_CONTENT

Next Generation Medical Infusion Pump Case Study

This case study summarizes the systems engineering aspects of the next-generation Symbiq™ IV (intravenous) medical pump development. Symbiq™ was developed by Hospira Inc. and documented in detail in Chapter 5 of the National Research Council book, *Human-System Integration in the System Development Process.* As described in the book, Symbiq™'s purpose was "to deliver liquid medications, nutrients, blood and other solutions at programmed flow rates, volumes and time intervals via intravenous and other routes to a patient, primarily for hospital use with secondary limited feature use by patients at home" (Pew 2007).

Domain Background

This case study provides insight into the use of systems engineering practices in a medical application.

Case Study Background

The project that is the subject of this report was approved by the Governing Board of the National Research Council, whose members are drawn from the councils of the National Academy of Sciences, the National Academy of Engineering, and the Institute of Medicine.

The study was supported by Award Nos. W911NF-05-0150 and FA5650-06-1-6610 between the National Academy of Sciences, the U.S. Department of the Army, and the U.S. Department of the Air Force.

Case Study Description

In creating a next-generation product, Hospira proposed to introduce new IV pump features, such as:

- multi-channel vs. single-channel liquid delivery;
- the ability to group multi-channeled devices together;
- associated user-programming capabilities and programmable drug libraries for specifying parallel delivery of liquids;
- use of color touchscreen devices;
- integration with numerous types of hospital information systems;
- ease of use for both medical personnel and patients at home;
- handling of potential hardware, software, and human-user faults;
- compliance with U.S. and international safety standards;
- use of alternating-current or battery power; and
- the ability to be cost-competitive and attractive to traditional medical and hospital administration personnel.

Many of these features are highly coupled, such as the multi-channel hardware controls, concurrent software synchronization, distinctive displays and alarms for multi-channel devices, and rigorous medical safety standards.

Views of the resulting medical infusion pump can be found as Figures 5-5 and 5-6 in Chapter 5, page 107 of the Pew and Mavor (2007) book. Systems engineering for the device involved a great deal of concurrent analysis and engineering of its hardware, software, human factors, operational, business, and safety aspects. It has been a commercial success and won the 2006 Human Factors and Ergonomics Society's User-Centered Product Design Award and the 2007 Medical Design Excellence Award.

Not only were there numerous technical challenges in the development of Symbiq™, but there were also challenges in the systems engineering of a product with a life-cycle operational concept that would produce satisfactory outcomes for a wide variety of product and operational stakeholders whose value propositions were often in some conflict. Some stakeholders wanted numerous features that would require a complex user interface, while others wanted a simple and easy to learn interface. Some users wanted the most advanced color touchscreen displays available, while others wanted a simpler, less-expensive product that was harder to misuse due to inadvertent screen commands. Some organizations felt that a minimal interpretation of the required safety features would be acceptable, while others advocated ultrahigh assurance levels. Some marketing personnel wanted a quick development and fielding of the basic product to capture market share, while maintainers wanted initial built-in life cycle support, maintenance, and diagnostic capabilities.

In such situations, many organizations focus on making quick requirement decisions and rapidly proceed into development. However, Hospira's understanding of the uncertainties and risks caused them to pursue a risk-driven, incremental commitment course of buying information to reduce risk, as emphasized in the SEBoK Part 3 knowledge area on Risk Management. As described in Pew and Mavor (2007), Hospira used a version of the Incremental Commitment Spiral Model (ICSM) summarized in the SEBoK Part 3 Knowledge Area on representative systems engineering process models. The following sections describe the project's incremental system definition progress through the ICSM exploration, valuation, foundations, and Development phases. Some evolution of terminology has occurred, the Pew and Mavor (2007) version uses ICM instead of ICSM and "architecting phase" instead of "foundations phase".

Symbiq™ Exploration Phase Summary

In the exploration phase, the project carried out numerous analyses on stakeholder needs, technical opportunities, and business competition. Using these analyses, the project team determined ranges of preferred options. Stakeholder needs analyses included contextual inquiry via shadowing of nurses using IV pumps and followup interviews, as well as creating task flow diagrams, use environment analyses, and user profiles analyses. Technical opportunity analyses included initial conceptual designs of multi-channel pump configurations, evaluation of commercially available single-color and multicolor display devices with touchscreen capabilities, and software approaches for specifying multi-channel delivery options and synchronizing concurrent processes.

Business competition analyses included hiring a management and marketing planning firm to examin next-generation pump competitor strengths and weaknesses with respect to such capabilities as the number of pump channels, therapies, programming options, air-in-line management, battery and alternating current capabilities, biomedical domain expertise, and alarms. Several key competitive advantages of a next-generation pump were identified, such as the ability to read bar-codes, small size, light weight, stand-alone functional channels, an extensive drug library, a high level of reliability, and clear mapping of screen displays and pumping channels.

Market research and market segment analyses also identified market windows, pricing alternatives, hospital purchasing decision-making trends, and safety aspects. These were iterated by focus groups of key thought leaders in critical care. The results were factored into a product concept plan, cost analysis, and business case analysis. These were independently reviewed by experts as part of the ICSM Valuation Phase Commitment Review process, which resulted in a go-ahead decision with an identification of several risks to be managed.

Symbiq™ Valuation Phase Summary

The valuation phase focused on the major risks highlighted in the Valuation Commitment Review, such as the multi-channel pump options, the types of programmable therapies, the need for tailorable medication libraries, the display screen and user interface options, and the safety considerations. The valuation phase also elaborated the product concept plan for the most attractive general set of options, including a development plan and operations plan, along with an associated cost analysis, risk analysis, and business case for review at the Foundations Commitment Review.

The multi-channel pump options were explored via several hardware industrial design mockups and early usability tests of the mockups. These included evaluation of such desired capabilities as semi-automatic cassette loading, special pole-mounting hardware, stacking of and total number of channels, and tubing management features. The evaluations led to the overall all choice to use a semi-automatic cassette loading capability with a red-yellow-green LED display to indicate concerns with the loading mechanism and with the pump in general.

Field exercises with prototypes of the pole mountings indicated the need for quick release and activation mechanisms, which were subsequently implemented. Risk analyses of alternative stacking mechanisms and the potential number of channels available established a preference for side-by-side stacking, a decision to develop one and two channel units, and to support a maximum of four channels in a stacked configuration.

The types of programmable therapies considered included continuous delivery for a specified time period, patient weight-based dosing, piggyback or alternating delivery between the two channels, tapered or ramped-rate delivery, intermittent-interval delivery, variable-time delivery, and multistep delivery. These were evaluated via prototyping of the software on a simulated version of the pump complexes and were iterated until satisfactory versions were found.

Evaluation of the tailorable medication libraries addressed the issue that different hard and soft safety limits were needed for dosages in different care settings (e.g., emergency room, intensive care, oncology, pediatric care, etc.) which creates a need for hospitals to program their own soft limits (overridable by nurses with permission codes) and hard limits (no overrides permitted). Stakeholder satisfaction with the tailoring features was achieved via prototype exercises and iteration with representative hospital personnel.

A literature review was conducted to determine the relative advantages and disadvantages of leading input and display technologies, including cost and reliability data. After down-selecting to three leading vendors of touch screen color LCD displays and further investigating their costs and capabilities, a business risk analysis focused on the trade offs between larger displays and customer interest in small-footprint IV pumps. The larger display was selected based on better readability features and the reduced risk of accidental user entries since the larger screen buttons would help to avoid these occurrences. Extensive usability prototyping was done with hardware mockups and embedded software that delivered simulated animated graphic user interface (GUI) displays to a touchscreen interface that was integrated into the hardware case.

The safety risk analysis in the valuation phase followed ISO 14971:2000 standards for medical device design, focusing on Failure Modes and Effects Analyses (FMEAs). This analysis was based on the early high-level design, such as entry of excessive drug doses or misuse of soft safety limit overrides. Subsequent-phase FMEAs would elaborate this analysis, based on the more detailed designs and implementations.

As in the exploration phase, the results of the valuation phase analyses, plans, budgets for the succeeding phases, the resulting revised business case, evidence of solution feasibility, and remaining risks with their risk management plans were reviewed by independent experts and the ICSM Foundations Commitment Review was passed, subject to a few risk level and risk management adjustments.

Symbiq™ Foundations Phase Summary

During the foundations phase, considerable effort was focused on addressing the identified risks such as the need for prototyping the full range of GUI usage by the full range of targeted users, including doctors, home patients, the need for interoperability of the Symbiq™ software with the wide variety of available hospital information systems, and the need for fully detailed FMEAs and other safety analyses. Comparable added effort went into detailed planning for development, production, operations, and support, providing more accurate inputs for business case analyses.

GUI prototyping was done with a set of usability objectives, such as

- 90% of experienced nurses will be able to insert the cassette the first time while receiving minimal training;
- 99% will be able to correct any insertion errors;
- 90% of first time users with no training will be able to power the pump off when directed; and
- 80% of patient users would rate the overall ease of use of the IV pump three or higher on a five-point scale (with five being the easiest to use).

Similar extensive evaluations were done on the efficacy and acceptability of the audio alarms, including the use of a patient and intensive care unit simulator that included other medical devices that produced noises, as well as other distractions such as ringing telephones. These evaluations were used to enable adjustment of the alarms and to make the visual displays easier to understand.

Software interoperability risk management involved extensive testing of representative interaction scenarios between the Symbiq™ software and a representative set of hospital information systems. These resulted in several adjustments to the software interoperability architecture. Also, as the product was being developed as a platform for the next generation of infusion pump products, the software design was analyzed for overspecialization to the initial product, resulting in several revisions. Similar analyses and revisions were performed for the hardware design.

As the design was refined into complete build-to specifications for the hardware and the operational software, the safety analyses were elaborated into complete FMEAs of the detailed designs. These picked up several potential safety issues, particularly involving the off-nominal usage scenarios, but overall confirmed a high assurance level for the safety of the product design. However, the safety risk assessment recommended a risk management plan for the development phase to include continued FMEAs, thorough off-nominal testing of the developing product’s hardware and software, and extensive beta-testing of the product at representative hospitals prior to a full release.

This plan and the other development and operations phase plans, product feasibility evidence, and business case analysis updates were reviewed at a Development Commitment Review, which resulted in a commitment to proceed into the development phase.

Symbiq™ Development Phase Systems Engineering Summary

The development phase was primarily concerned with building and testing the hardware and software to the build-to specifications, but continued to have an active systems engineering function to support change management; operations, production, and support planning and preparation; and further safety assurance activities as recommended in the risk management plan for the phase.

For hospital beta-testing, thoroughly bench-tested and working beta versions of the IV pump were deployed in two hospital settings. The hospitals programmed drug libraries for at least two clinical care areas. The devices were used for about four weeks. Surveys and interviews were conducted with the users to capture their “real world” experiences with the pump. Data from the pump usage and interaction memory was also analyzed and compared to the original doctors’ orders. The beta tests revealed a number of opportunities to make improvements, including revision of the more annoying alarm melodies and the data entry methods for entering units of medication delivery time in hours or minutes.

Usability testing was also conducted on one of the sets of abbreviated instructions called TIPS cards. These cards serve as reminders for how to complete the most critical tasks. Numerous suggestions for improvement in the TIPS cards themselves, as well as the user interface, came from this work, including how to reset the “Air-in-Line” alarm and how to check all on-screen help text for accuracy.

The above mentioned usability objectives were used as Acceptance Criteria (glossary) for the validation usability tests. These objectives were met. For example, the calculated task completion accuracy was 99.66% for all tasks for first time nurse users with minimal training. There were a few minor usability problems uncovered that were subsequently fixed without major changes to the GUI or effects on critical safety related tasks.

The risk analysis was iterated and revised as the product development matured. FMEAs were updated for safety critical risks associated with three product areas: the user interface, the mechanical and electrical subsystems, and the product manufacturing process. Some detailed implementation problems were found and fixed, but overall the risk of continuing into full-scale production, operations, and support was minimal. Systems engineering continued into the operations phase, primarily to address customer change requests and problem reports, and to participate in planning for a broader product line of IV pumps.

Overall, customer satisfaction, sales, and profits from the Symbiq™ IV pump have been strong and satisfaction levels from the management, financial, customer, end user, developer, maintainer, regulatory, and medical-community stakeholders have been quite high (Pew 2007).

Summary

In summary, the Symbiq™ Medical Infusion Pump Case Study provides an example of the use of the systems engineering practices discussed in the SEBoK. As appropriate for a next-generation, advanced technology product, it has a strong focus on risk management, but also illustrates the principles of synthesis, holism, dynamic behavior, adaptiveness, systems approach, progressive entropy reduction, and progressive stakeholder satisfying discussed in Part 2 of the SEBoK. It provides an example of an evolutionary and concurrent systems engineering process, such as the incremental commitment spiral process, and of other knowledge areas discussed in SEBoK Parts 3 and 4, such as system definition, system realization, system engineering management, and specialty engineering.

References

Works Cited

Pew, R. and A. Mavor. 2007. *Human-System Integration in the System Development Process: A New Look.* Washington, DC, USA: The National Academies Press.

Primary References

None.

Additional References

None.

< Previous Article | Parent Article | Next Article >

SEBoK v. 1.4 released 29 June 2015

SEBoK Discussion

Please provide your comments and feedback on the SEBoK below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account. Simply type your comment in the text field below and DISQUS will guide you through the login or registration steps. Feedback will be archived and used for future updates to the SEBoK. *If you provided a comment that is no longer listed, that comment has been adjudicated. You can view adjudication for comments submitted prior to SEBoK v. 1.0 at SEBoK Review and Adjudication. Later comments are addressed and changes are summarized in the Letter from the Editor and Acknowledgements and Release History.*

If you would like to provide edits on this article, recommend new content, or make comments on the SEBoK as a whole, please see the SEBoK Sandbox [1].

ENCODED_CONTENT
MTgyMTQPGRpdiBpZD0iZGlzcXVzX3RocmVhZCI+PC9kaXY+CjxzY3JpcHQgdHlwZT0idGV4dC9qYXZhc2Nya
END_ENCODED_CONTENT

Design for Maintainability

This article describes an example of where systems thinking led to a much more practical solution to a common problem. For addition information, refer to Systems Thinking.

This article is excerpted and condensed from Johnson, Steven. 2010. *Where Good Ideas Come From: The Natural History of Innovation*. New York: Riverhead Books. pp. 25-28.

Background

In the late 1870s a Parisian obstetrician named Stephane Tarnier was visiting the Paris Zoo where they had farm animals. While there he conceived the idea of adapting a chicken incubator to use for human newborns, and he hired “the zoo’s poultry raiser to construct a device that would perform a similar function for human newborns.” At the time infant mortality was staggeringly high “even in a city as sophisticated as Paris. One in five babies died before learning to crawl, and the odds were far worse for premature babies born with low birth weights.” Tarnier installed his incubator for newborns at Maternité de Paris, and embarked on a quick study of five hundred babies. “The results shocked the Parisian medical establishment: while 66 percent of low-weight babies died within weeks of birth, only 38 percent died if they were housed in Tarnier’s incubating box. … Tarnier’s statistical analysis gave newborn incubation the push that it needed: within a few years the Paris municipal board required that incubators be installed in all the city’s maternity hospitals.” …

Purpose

“Modern incubators, supplemented with high-oxygen therapy and other advances, became standard equipment in all American hospitals after the end on World War II, triggering a spectacular 75 percent decline in infant mortality rates between 1950 and 1998.”… “In the developing world, however, the infant mortality story remains bleak. Whereas infant deaths are below ten per thousand births throughout Europe and the United States, over a hundred infants die per thousand (births) in countries like Liberia and Ethiopia, many of them premature babies that would have survived with access to incubators.

Challenges

But modern incubators are complex, expensive things. A standard incubator in an American hospital might cost more than $40,000 [about €30,000]. But the expense is arguably the smaller hurdle to overcome. Complex equipment breaks and when it breaks you need the technical expertise to fix it, and you need replacement parts. In the year that followed the 2004 Indian Ocean tsunami, the Indonesian city of Meulaboh received eight incubators from a range on international relief organizations. By late 2008, when an MIT professor named Timothy Prestero visited the hospital, all eight were out of order, the victims of power surges and tropical humidity, along with the hospital staff's inability to read the English repair manual. The Meulaboh incubators were a representative sample: some studies suggest that as much as 95 percent of medical technology donated to developing countries breaks within the first five years of use.

Systems Engineering Practices

"Prestero had a vested interest in those broken incubators, because the organization he founded, Design that Matters, had been working for several years on a scheme for a more reliable, and less expensive, incubator, one that recognized complex medical technology was likely to have a very different tenure in a developing world context than it would in an American or European hospital. Designing an incubator for a developing country wasn't just a matter of creating something that worked; it was also a matter of designing something that would break in a non-catastrophic way. You couldn't guarantee a steady supply of spare parts, or trained repair technicians. So instead, Prestero and his team decided to build an incubator out of parts that were already abundant in the developing world. The idea had originated with a Boston doctor named Jonathan Rosen, who had observed that even the smaller towns of the developing world seemed to be able to keep automobiles in working order. The towns might lack air conditioning and laptops and cable television, but they managed to keep their Toyota 4Runners on the road. So Rosen approached Prestero with an idea: What if you made an incubator out of automobile parts?

Lessons Learned

"Three years after Rosen suggested the idea, the Design that Matters team introduced a prototype device called NeoNurture. From the outside, it looked like a streamlined modern incubator, but its guts were automotive. Sealed-beam headlights supplied the crucial warmth; dashboard fans provided filtered air circulation; door chimes sounded alarms. You could power the device via an adapted cigarette lighter, or a standard-issue motorcycle battery. Building the NeoNurture out of car parts was doubly efficient, because it tapped both the local supply of parts themselves and the local knowledge of automobile repair. These were both abundant resources in the developing world context, as Rosen liked to say. You didn't have to be a trained medical technician to fix the NeoNurture; you didn't even have to read the manual. You just needed to know how to replace a broken headlight."

References

Works Cited

Johnson, Steven. 2010. Where Good Ideas Come From: The Natural History of Innovation. New York: Riverhead Books. pp. 25-28.

Primary References

Johnson, Steven. 2010. Where Good Ideas Come From: The Natural History of Innovation. New York: Riverhead Books. pp. 25-28.

Additional References

None.

< Previous Article | Parent Article | Next Article >

SEBoK v. 1.4 released 29 June 2015

SEBoK Discussion

Please provide your comments and feedback on the SEBoK below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account. Simply type your comment in the text field below and DISQUS will guide you through the login or registration steps. Feedback will be archived and used for future updates to the SEBoK. *If you provided a comment that is no longer listed, that comment has been adjudicated. You can view adjudication for comments submitted prior to SEBoK v. 1.0 at SEBoK Review and Adjudication. Later comments are addressed and changes are summarized in the Letter from the Editor and Acknowledgements and Release History.*

If you would like to provide edits on this article, recommend new content, or make comments on the SEBoK as a whole, please see the SEBoK Sandbox [1].

ENCODED_CONTENT
NTc4NTEPGRpdiBpZD0iZGlzcXVzX3RocmVhZCI+PC9kaXY+CjxzY3JpcHQgdHlwZT0idGV4dC9qYXZhc2NyaX
END_ENCODED_CONTENT

Complex Adaptive Operating System Case Study

This article revolves around project management in complex adaptive systems and focuses on creating tool and methods that project managers can use in managing complex projects.

Background

The International Centre for Complex Project Management (ICCPM) is a non-profit organization that was created to address “the international community’s ability to successfully deliver very complex projects and manage complexity across all industry and government sectors” (ICCPM, 2012).

In an ongoing effort to help member organizations successfully undertake major complex projects, ICCPM conducted a bi-annual series of international roundtables. The purpose of the roundtables was to better understand what contributes to the success of complex projects and to identify and develop new and improved tools and approaches. The roundtables were facilitated using a computer-assisted collaborated meeting process that leverages the features of complex adaptive systems—described below—to help people with diverse viewpoints and experience create new collective understanding and plans for action.

Complex major projects are known for being unsuccessful in on-time and on-budget completion. An (IBM, 2008) survey of 1,500 change managers found that only 40% of projects finished on time and on budget. Barriers to success were the inability to change attitudes or mindsets (58%), dysfunctional culture (40%) lack of senior management support (32%) and underestimating the complexity of a project (35%).

However, several new systemic approaches show considerable promise as a way to think about and manage projects. Six frameworks help inform these approaches: systems thinking, the features of complex adaptive systems (CASs), complexity theory, the Complexity Model of Change (Findlay and Straus, 20011); polarity thinking as a way of thinking about and leveraging wicked problems, cognitive complexity and adult development theory.

Systems thinking recognizes whole systems and the interdependencies of their parts. A system may be defined as “a set of things, organisms, and people that, as a result of their interconnection, produce their own patterns of behavior over time” (Meadows, 2008, p.2). A system cannot be understood by focusing on its parts alone (Wheatley, 1999). To successfully address and influence a system, such as a complex project, one must understand the whole and how its parts interact.

Some project managers continue to think of a project as a “machine” that operates according to linear or algorithm rules. This persistent and largely unhelpful meme is now being displaced by a new and more robust model, which regards projects as complex adaptive systems (CASs). Unlike strictly mechanical systems, CASs are self-organizing, learn from experience, are emergent, and from time to time undergo large scale phase transitions to new and higher states of the system. This view of large-scale projects—which are heavily influenced by human interaction—is proving to be much more accurate and allows project managers to leverage techniques for exerting influence in environments that have previously presented intractable problems .

Leaders of complex projects would also be wise to consider three fundamental theorems of complexity theory, which apply to CASs and which are critical to project success. These are a robust model of the system, requisite variety and adopting solutions which act at an appropriate scale.

- The robust model axiom considers that “no one can effectively influence a system until they have a thorough understanding of its scope and the connections and interdependencies” (Conant and Ashby, 1970).
- The law of requisite variety contents that “complexity can only be dealt with by equal or greater complexity” (Ashby, 1956, p. 2). In other words, in order to deal effectively with the diversity of problems presented by a complex project, one must have a repertoire of responses which is (at least) as nuanced as the problems themselves (Requisite Variety, 2015).

- The scale condition requires that those who wish to exercise leverage over a system must recognize that “highly complex situations can best be addressed by greater degrees of freedom at the local scale so that innovation and adaptability are maximized” (Bar-Yam, 2004).

A third framework, the Complexity Model of Change, is a model of socio-technological change, comprising a series of growth and decay curves or waves, which helps project managers better understand how to influence systems and design new ones, so the roles, methods, rules of interaction or engagement, technologies and relationships between people are better aligned with each other and the desired outcome. The overlapping waves represent large-scale eras, for example, the Industrial Age and the Information Age, which have at their core a metaphor, for example the machine and the computer. The current wave, the Knowledge Age, is based on a network metaphor. The wave we are now entering is the Wisdom Age, which began in 2010. Its core metaphor is the complex adaptive system and the main thrust of this period is the wise application of knowledge.

Another area that project managers may now address differently is that of wicked problems or paradoxes: problems that are ill-defined and recurrent, and which, when attempts are made to solve them with single optimal solutions, create another problem. Polarity thinking regards wicked problems as sets of interdependent values or ideas—like centralizing for efficiency and decentralizing for adaptability—that persist together over time and need each other for the success of the system. If we pay attention to one pole at the expense of the other we achieve sub-optimal results. When we manage polarities as a system, we realize the benefits of both poles and achieve high performance over time with a minimum of vacillation and the need for correction.

Other disciplines that are critical to project success are understanding and making best use of new ways of thinking about issues and relating to others in more flexible and adaptive ways. Theories of cognitive complexity and adult development theory can contribute to how we think about this problem. For example, “triple-loop learning” (Gragert, 2013), helps us think about issues from a higher level of cognitive complexity. Instead of asking are we doing things right (single loop), we might ask are we doing the right things (double loop) or how do we decide what is the most effective paradigm to use to influence and create benefit for the system (triple)?

Purpose

The purpose of the ICCPM roundtables was to help project managers develop new and better ways to lead complex major projects, by bringing together people from both the buy-side and the supply-side to share their knowledge and experience and to grow a network of practitioners, professionals, researchers, and educators able to deliver leading edge complex project management solutions to client organizations and partners around the world (Findlay and Straus, 2015, p. 489).

Challenges

There are many challenges to be addressed in the complex project management environments. The three top contenders are 1) developing new ways of thinking, acting, and interacting; 2) developing more robust models of the system by getting everyone in the room—the project management team and their stakeholders; and 3) steering projects through multiple disruptive socio-technological shifts using the feature of complex adaptive systems.

```
“People, their organizations, and their projects need to be capable of
reorganizing into new forms, which are a better fit with the new
context” (Findlay and Straus, 2015, p. 494).
```

Systems Engineering Practices

One of the tools Findlay and Straus use to deal with all three challenges in the context of group interaction, such as the ICCPM roundtables, is the Zing complex adaptive meeting process. The process is used to guide conversation in the room, to capture, simultaneously display ideas and to help participants integrate and make meaning from the ideas. The tool was used for the roundtables to help people work together in new ways, develop new and better models of the system together and to design and pilot new and better decision and learning methods.

The technology "provide[s] a container for a suitably representative sample of the people in the system to meet and conceptualize a robust model of the system and develop strategies for how to leverage the system" (Findlay and Straus, 2015, p. 492).

A "talk-type-read-review" (Findlay and Straus, 2015, p. 492) etiquette was employed to organize the session, which, in complexity theory terms, is a simple rule of interaction. Rich, open-ended questions guide the conversations, the ideas are read out aloud and the common themes or stand out ideas are recorded by the facilitators.

The open-ended questions are asked one at a time to explore all possibilities and reduce complexity. Although, roundtable participants often held opposing views at the beginning, of the session, through a processes of continual, iterative feedback, they ultimately arrived at similar or complementary conclusions by the end of the roundtable.

The process "automates"—or helps participants engage in—ways of interacting that incorporate a higher level of cognitive complexity than the participants might engage in individually, thus facilitating a shift in the group to a higher level of system performance.

Lessons Learned

The role of leaders of complex projects is to help their organization systems successfully deliver on time and on budget amidst constant change. Their mandate is to deliver amazing new solutions while making few or no mistakes—a challenging goal even in far less complex environments. In order to be successful, project leaders (and their teams) need new systems structures—new tools and methods—that reliably get better results. They need to have a robust and fresh understanding of the systems over which they preside and how they might influence them to greatest effect.

No longer can the complex project leader go off into a corner and design a project and then try to sell it to the community and political leaders, for example. Leaders now need to involve the whole system in the design of a project from its inception through to completion. They need to deal with wicked problems not by looking for the one best solution, but by integrating and leveraging competing ideas. This requires a shift in perspective: from attempting to "control" a complex project system as one might control a mechanical device, to understanding projects as highly complex and interconnected "living" systems that evolve over time. While we do not have "control" over our systems in the classical sense, we can exert influence very effectively, provided that we constantly update our understanding of what is going on and learn new ways to act and interact that are more likely to achieve our desired outcomes.

To achieve this, leaders need to develop the capacity to "anticipate the skills, leadership and coordination roles, technologies, methods, and processes that will be required to successfully surf the waves of change…" (Findlay and Straus, 2015, p. 501).

The 2012 ICCPM roundtable series discussion paper (ICCPM, 2012) uses the example of a system undergoing transformation of many levels to illustrate the difficulty that complex project leaders face:

```
"The issue has been characterized as learning to fly a plane, while the
 plane is already in the air, and being re-assembled into another kind
of transportation technology altogether. And, at the same time, the
current passengers are disembarking and another group is boarding that
demands a better quality of service or experience at lower cost than
```

```
ever before. (ICCPM 2015 p. 21)”
```

This case study illustrates the need, in times of accelerating change, of “a real-time, systems-wide approach to the development of the methods and tools for managing complex projects” (Findlay and Straus, 2015, p. 500) so leaders can deal successfully and creatively with uncertainty and ambiguity.

References

Works Cited

Ashby, R. 1956. *An introduction to cybernetics*. London: Chapman and Hall.

Bar Yam, Y. 2004. *Making things work: Solving complex problems in a complex world*. Cambridge, MA: Knowledge Press.

Conant, R., and Ashby, R. 1970. *Every good regulator of a system must be a model of that system*. International Journal of System Sciences. 1(2): 89-97.

Findlay, J., and Straus, A. 2011. *A shift from systems to complex adaptive systems thinking*. In O. Bodrova and N. Mallory (Eds.), Complex Project Management Task Force Report: Compendium of Working Papers. Canberra: International Centre for Complex Project Management: 24-26.

Findlay, J., and Straus, A. 2015. *Complex Adaptive Operating System: Creating Methods for Complex Project Management*. In Case Studies in System of Systems, Enterprise Systems, and Complex Systems Engineering. Gorod, A., White, B.E., Ireland, V., Gandhi, S.J., and Sauser, B. (Eds.). Boca Raton: CRC Press: 471-505.

Gragert, T. 2013. *Triple loop learning*. Thorston’s site. Retrieved 12 June, 2015 from http://www.thorsten.org/wiki/index.php?title=Triple_Loop_Learning.

IBM. 2008 Making Change Work. New York: IBM.

ICCPM. 2012. *Roundtable discussion paper: Complexity in a time of global financial change: Program delivery for the new economy*. Findlay, J., Straus, A., and Hatcher, C. (Eds). Canberra: International Centre for Complex Project Management.

Meadows, D. 2008. *Thinking in Systems: A primer*. White River Junction, Vermont: Chelsea Green Publishing.

Requisite Variety. 2015. *What is requisite variety?*. Retrieved 1 June, 2015 from http://requisitevariety.co.uk/what-is-requisite-variety.

Wheatley, M. 1999. *Leadership and the new science: Discovering order in a chaotic world*. San Francisco: Berrett-Koehler.

Primary References

Findlay, J., and Straus, A. 2015. *Complex Adaptive Operating System: Creating Methods for Complex Project Management*. In Case Studies in System of Systems, Enterprise Systems, and Complex Systems Engineering. Gorod, A., White, B.E., Ireland, V., Gandhi, S.J., and Sauser, B. (Eds.). Boca Raton: CRC Press: 471-505.

Additional References

None.

< Previous Article | Parent Article | Next Article>

SEBoK v. 1.4 released 29 June 2015

SEBoK Discussion

Please provide your comments and feedback on the SEBoK below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account. Simply type your comment in the text field below and DISQUS will guide you through the login or registration steps. Feedback will be archived and used for future updates to the SEBoK. *If you provided a comment that is no longer listed, that comment has been adjudicated. You can view adjudication for comments submitted prior to SEBoK v. 1.0 at SEBoK Review and Adjudication. Later comments are addressed and changes are summarized in the Letter from the Editor and Acknowledgements and Release History.*

If you would like to provide edits on this article, recommend new content, or make comments on the SEBoK as a whole, please see the SEBoK Sandbox [1].

ENCODED_CONTENT
MTg1MDgPGRpdiBpZD0iZGlzcXVzX3RocmVhZCI+PC9kaXY+CjxzY3JpcHQgdHlwZT0idGV4dC9qYXZhc2Nya
END_ENCODED_CONTENT

Vignettes

Systems engineering (SE) principles described in the SEBoK Parts 1-6 are illustrated in Part 7, Systems Engineering Implementation Examples. These examples describe the application of systems engineering practices, principles, and concepts in real settings. These examples can be used to improve the practice of systems engineering by illustrating to students and practitioners the benefits of effective practice and the risks of poor practice.

The SEBoK systems engineering implementation examples are grouped in two categories: case studies and vignettes. The SEBoK examines case studies previously published by external sources and demonstrates the real world examples of systems engineering principles that are present in these studies. The vignettes are short wiki articles written specifically for the SEBoK. These vignettes were developed to illustrate the applicability of systems engineering principles in a broader range of domains and geographic locations.

A matrix is used to map the implementation examples to topics in the SEBoK. This matrix maps each implementation example to the discussion of the specific systems engineering principles illustrated.

List of Vignettes

The following vignettes are included:

- Denver Airport Baggage Handling System Vignette
- Virginia Class Submarine Vignette
- UK West Coast Route Modernisation Project Vignette
- Singapore Water Management Vignette
- FAA Advanced Automation System (AAS) Vignette
- Standard Korean Light Transit System Vignette

References

None.

< Previous Article | Parent Article | Next Article >

SEBoK v. 1.4 released 29 June 2015

SEBoK Discussion

Please provide your comments and feedback on the SEBoK below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account. Simply type your comment in the text field below and DISQUS will guide you through the login or registration steps. Feedback will be archived and used for future updates to the SEBoK. *If you provided a comment that is no longer listed, that comment has been adjudicated. You can view adjudication for comments submitted prior to SEBoK v. 1.0 at SEBoK Review and Adjudication. Later comments are addressed and changes are summarized in the Letter from the Editor and Acknowledgements and Release History.*

If you would like to provide edits on this article, recommend new content, or make comments on the SEBoK as a whole, please see the SEBoK Sandbox [1].

ENCODED_CONTENT
MTgyMTQPGRpdiBpZD0iZGlzcXVzX3RocmVhZCI+PC9kaXY+CjxzY3JpcHQgdHlwZT0idGV4dC9qYXZhc2Nya
END_ENCODED_CONTENT

Denver Airport Baggage Handling System Vignette

This vignette describes systems engineering (SE) issues related to the development of the automated baggage handling system for the Denver International Airport (DIA) from 1990 to 1995. The computer controlled, electrical-mechanical system was part of a larger airport system.

Vignette Description

In February 1995, DIA was opened 16 months later than originally anticipated with a delay cost of $500 million (Calleam Consulting Ltd. 2008). A key schedule and cost problem—the integrated automated baggage handling system—was a unique feature of the airport. The baggage system was designed to distribute all baggage automatically between check-in and pick-up on arrival. The delivery mechanism consisted of 17 miles of track on which 4,000 individual, radio-controlled carts would circulate. The $238 million system consisted of over 100 computers networked together, 5,000 electric eyes, 400 radio receivers, and 56 bar-code scanners. The purpose of the system was to ensure the safe and timely arrival of every piece of baggage. Significant management, mechanical, and software problems plagued the automated baggage handling system. In August 2005, the automated system was abandoned and replaced with a manual one.

The automated baggage system was far more complex than previous systems. As planned, it would have been ten times larger than any other automated system, developed on an ambitious schedule, utilized novel technology, and required shorter-than-average baggage delivery times. As such, the system involved a very high level of SE risk. A fixed scope, schedule, and budget arrangement precluded extensive simulation or physical testing of the full design. System design began late as it did not begin until well after construction of the airport was underway. The change management system allowed acceptance of change requests that required significant redesigns to portions of work already completed. The design did not include a meaningful backup system; for a system that required very high mechanical and computer reliability, this increased failure risks. The system had an insufficient number of tugs and carts to cope with the volume of baggage expected and this, along with severely limited timing requirements, caused baggage carts to jam in the tracks and for them to misalign with the conveyor belts feeding the bags. This resulted in mutilated and lost bags (Neufville 1994; Gibbs 1994).

The baggage system problems could be associated with the non-use or misuse of a number of systems engineering (SE) concepts and practices: system architecture complexity, project scheduling, risk management, change management, system analysis and design, system reliability, systems integration, system verification and validation/testing, and insufficient management oversight.

Summary

The initial planning decisions, such as the decision to implement one airport-wide integrated system, the contractual commitments to scope, schedule, and cost, as well as the lack of adequate project management (PM) procedures and processes, led to a failed system. Attention to SE principles and practices might have avoided the system's failure.

References

Works Cited

Calleam Consulting Ltd. 2008. *Case Study – Denver International Airport Baggage Handling System – An illustration of ineffectual decision making*. Accessed on September 11, 2011. Available at http://calleam.com/WTPF/?page_id=2086.

Neufville, R. de. 1994. "The Baggage System at Denver: Prospects and Lessons." *Journal of Air Transport Management*. 1(4): 229-236.

Gibbs, W.W. 1994. "Software's Chronic Crisis." *Scientific American*. September 1994: p. 72-81.

Primary References

None.

Additional References

DOT. 1994. "New Denver Airport: Impact of the Delayed Baggage System." US Department of Transportation (DOT), Research Innovation Technology Administration. GAO/RCED-95-35BR. Available at http://ntl.bts.gov/DOCS/rc9535br.html

Donaldson, A.J.M. 2002. "A Case Narrative of the Project Problems with the Denver Airport Baggage Handling Systems (DABHS)," Software Forensics Center technical Report TR 2002-01. Middlesex University, School of Computer Sciences. Available at http://www.eis.mdx.ac.uk/research/SFC/Reports/TR2002-01.pdf

< Previous Article | Parent Article | Next Article >

SEBoK v. 1.4 released 29 June 2015

SEBoK Discussion

Please provide your comments and feedback on the SEBoK below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account. Simply type your comment in the text field below and DISQUS will guide you through the login or registration steps. Feedback will be archived and used for future updates to the SEBoK. *If you provided a comment that is no longer listed, that comment has been adjudicated. You can view adjudication for comments submitted prior to SEBoK v. 1.0 at SEBoK Review and Adjudication. Later comments are addressed and changes are summarized in the Letter from the Editor and*

Acknowledgements and Release History.

If you would like to provide edits on this article, recommend new content, or make comments on the SEBoK as a whole, please see the SEBoK Sandbox [1].

ENCODED_CONTENT
NTQzMDgPGRpdiBpZD0iZGlzcXVzX3RocmVhZCI+PC9kaXY+CjxzY3JpcHQgdHlwZT0idGV4dC9qYXZhc2Nya
END_ENCODED_CONTENT

Virginia Class Submarine Vignette

Prior to the Virginia class submarine, sonar systems were comprised of proprietary components and interfaces. However, in the mid-1990s the United States government transitioned to the use of commercially developed products - or commercial off the shelf (COTS) products - as a cost-saving measure to reduce the escalating costs associated with proprietary-based research and development. The Virginia class submarine system design represented a transition to COTS-based parts and initiated a global change in architectural approaches adopted by the sonar community. The lead ship of the program, Virginia, reduced the number of historically procured parts for nuclear submarines by 60% with the use of standardization. The Virginia class submarine sonar system architecture has improved modularity, commonality, standardization, and reliability, maintainability and testability (RMT) over historical sonar systems.

Architectural Approach: Standardization

Based on the new architectural approach and the success of the transition, system architecture experts developed an initial set of architecture evaluation metrics:

- Commonality
  - Physical commonality (within the system)
    - Hardware (HW) commonality (e.g., the number of unique line replaceable units, fasteners, cables, and unique standards implemented)
    - Software (SW) commonality (e.g., the number of unique SW packages implemented, languages, compilers, average SW instantiations, and unique standards implemented)
  - Physical familiarity (with other systems)
    - Percentage of vendors and subcontractors known
    - Percentage of HW and SW technology known
  - Operational commonality
    - Percentage of operational functions which are automated
    - Number of unique skill codes required
    - Estimated operational training time (e.g., initial and refresh from previous system)
    - Estimated maintenance training time (e.g., initial and refresh from previous system)
- Modularity
  - Physical modularity (e.g., ease of system element or operating system upgrade)
  - Functional modularity (e.g., ease of adding new functionality or upgrading existing functionality)
  - Orthogonality
    - Level to which functional requirements are fragmented across multiple processing elements and interfaces
    - Level to which throughput requirements span across interfaces
    - Level to which common specifications are identified
  - Abstraction (i.e., the level to which the system architecture provides an option for information hiding)

    - Interfaces
        - Number of unique interfaces per system element
        - Number of different networking protocols
        - Explicit versus implicit interfaces
        - Level to which the architecture includes implicit interfaces
        - Number of cables in the system
- Standards-based openness
    - Interface standards
        - Ratio of the number of interface standards to the number of interfaces
        - Number of vendors for products based on standards
        - Number of business domains that apply/use the standard (e.g., aerospace, medical, and telecommunications)
        - Standard maturity
    - Hardware standards
        - Ratio of the number of form factors to the number of line replaceable units (LRUs)
        - Number of vendors for products based on standards
        - Standard maturity
    - Software standards
        - Number of proprietary and unique operating systems
        - Number of non-standard databases
        - Number of proprietary middle-ware
        - Number of non-standard languages
    - Consistency orientation
        - Common guidelines for implementing diagnostics and performance monitor/fault location (PM/FL)
        - Common guidelines for implementing human-machine interface (HMI)
- Reliability, maintainability, and testability
    - Reliability (fault tolerance)
    - Critical points of fragility (e.g., system loading comprised of percent of processor, memory, and network loading)
    - Maintainability (e.g., expected mean time to repair (MTTR), maximum fault group size, whether the system can be operational during maintenance)
    - Accessibility (e.g., space restrictions, special tool requirements, special skill requirements)
    - Testability
        - Number of LRUs covered by built-in tests (BIT) (BIT coverage)
        - Reproducibility of errors
        - Logging/recording capability
        - Whether the system state at time of system failure can be recreated
        - Online testing (e.g., whether the system is operational during external testing and the ease of access to external test points)
        - Automated input/stimulation insertion

Other Points

The Virginia class submarine acquisition exhibited other best practices. These are discussed by Schank (2011), GAO (2008), and General Dynamics (2002).

These best practices included stringent design trades to keep costs under control, careful consideration of technical maturity of components, and the importance of program stability.

Summary

In summary, the work on the Virginia class submarine prompted a change in the traditional architectural approach used in the sonar community to design submarine sonar and validated the cost savings in both research and development (R&D) and in component costs when transitioning from proprietary interfaces to industry standard interfaces. The identification of a list of feasible architecture evaluation metrics was an added benefit of the effort.

References

Works Cited

GAO. 2008. *Defense Acquisitions: Assessment of Selected Weapon Programs Report*. Washington, DC, USA: US. Government Accountability Office (GAO). March 2009. GAO-09-326SP.

GD Electric Boat Division. 2002. *The Virginia Class Submarine Program: A Case Study*. Groton, CT: General Dynamics. February, 2002.

Schank, J.F. et al. 2011. *Learning from Experience, Volume 2: Lessons from the U.S. Navy's Ohio, Seawolf, and Virginia Submarine Programs.* Santa Monica, CA, USA: Rand. Available at http://www.rand.org/content/dam/rand/pubs/monographs/2011/RAND_MG1128.2.pdf [1]

Primary References

None.

Additional References

None.

< Previous Article | Parent Article | Next Article >

SEBoK v. 1.4 released 29 June 2015

SEBoK Discussion

Please provide your comments and feedback on the SEBoK below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account. Simply type your comment in the text field below and DISQUS will guide you through the login or registration steps. Feedback will be archived and used for future updates to the SEBoK. *If you provided a comment that is no longer listed, that comment has been adjudicated. You can view adjudication for comments submitted prior to SEBoK v. 1.0 at SEBoK Review and Adjudication. Later comments are addressed and changes are summarized in the Letter from the Editor and Acknowledgements and Release History.*

If you would like to provide edits on this article, recommend new content, or make comments on the SEBoK as a whole, please see the SEBoK Sandbox [1].

ENCODED_CONTENT
MjU5ODUPGRpdiBpZD0iZGlzcXVzX3RocmVhZCI+PC9kaXY+CjxzY3JpcHQgdHlwZT0idGV4dC9qYXZhc2Nya

END_ENCODED_CONTENT

References

[1] http://www.rand.org/content/dam/rand/pubs/monographs/2011/RAND_MG1128.2.pdf

UK West Coast Route Modernisation Project Vignette

The West Coast Main Line (WCML) is a principal United Kingdom (UK) railway artery serving London, the Midlands, the North West and Scotland. The Line is responsible for over 2,000 train movements each day, with more than 75 million rail journeys made each year on the route. It accounts for 43% of Britain's UK freight market (Railway People 2011). In 1998, the British government embarked on a modernization program called the West Coast Route Modernisation (WCRM) project, to carry out a significant volume of modernization work between 1998 and 2008, delivering increased capacity and reduced journey times as well as replacing worn-out parts of the railway. It was challenging a job involving 640 kilometers of track—much of which was incapable of carrying high-speed rail cars. Some sections were seriously dilapidated, and new trains would require a complete overhaul of signaling, power supply, and switching systems.

This vignette is based on information from an INCOSE publication on systems engineering case studies (INCOSE 2011) and a report of the UK National Audit Office (NAO 2006).

Vignette Description

Early on, the WCRM upgrade had serious problems. A major complicating factor was the introduction of a new signaling technology that was designed to allow improved services for new trains running at 140 miles per hour. By 2001, neither the rail infrastructure upgrade nor the new trains were on course for delivery as expected in the 1998 agreement. By May 2002 the projection of the program's final cost had risen from £2.5 billion (in 1998) to £14.5 billion, but had delivered only a sixth of the original scope.

In January 2002, the UK Secretary of State instructed the Strategic Rail Authority (SRA) to intervene and find a way to renew and upgrade the WCML. An SRA analysis identified the following issues:

- The program lacked direction and leadership before 2002.
- The project did not have a delivery strategy and there was no central point for responsibility and communication.
- There was a lack of openness and communication regarding the program with interested parties before 2002 and a lack of stakeholder management.
- Scope changes arose because WCRM did not have an agreed-upon specification that matched required outputs with inputs.
- There was inadequate knowledge about the West Coast asset condition.
- Technology issues related to the decision to replace conventional signaling with unproven moving block signaling introduced major risk into deliverability and cost before 2002. These technology issues caused scope changes and program delay.
- Project management (PM) was weak, with a lack of senior management skills, too many changes in personnel, and ill-defined and fragmented roles and responsibilities. There was no integrated delivery plan and there was limited oversight of contractors. Poor management of contracts added to costs.

In order to remedy the situation the SRA initiated the following actions, which align with generally accepted systems engineering (SE) practice:

- A clear direction for the project was developed and documented in the June 2003 West Coast Main Line Strategy, specifying desired goals and outcomes.
- A clear, measurable set of program outputs was established, along with more detailed infrastructure requirements, which were then subject to systematic change control and monitoring procedures fixing scope. Contractors were invited to tender complete detailed designs and deliver the work to a fixed price.
- Clear program governance structures were instituted.
- The SRA consulted widely with stakeholders and, in turn, kept stakeholders informed.

A National Audit Office (NAO) report concluded that the new arrangements worked well and that there were benefits to this approach. (NAO 2006) Until this time, one of the program's key constraints and cost drivers had been the ability to access certain areas of the track. The new approach facilitated the ability to obtain possession of the track for engineering work, which was crucial to delivery. The new approach also enabled the program to identify opportunities to reduce the total cost by over £4 billion.

The NAO report also discussed a business case analysis by the SRA that identified the following benefits (NAO 2006):

- benefit:cost ratio for the enhancements element was 2.5:1;
- journey times and train frequencies exceeded the targets set out in the 2003 West Coast Strategy;
- growth in passenger numbers exceeded expectations (e.g., by 2005-06, following Phase 1 of the West Coast program, annual passenger journeys on Virgin West Coast grew by more than 20%); and
- punctuality improved (e.g., by September 2006, average time delays on Virgin West Coast trains have been approximately 9.5 minutes, a 43% improvement on the average delay of 17 minutes in September 2004).

The WCRM problems could be associated with a number of systems engineering concepts and practices: stakeholders requirements, planning, analysis of risks and challenges of new technology and associated risk management, decision management, configuration or change management, information management, and management oversight.

Summary

The WCRM project illustrates that when SE concepts and practices are not used or applied properly, system development can experience debilitating problems. This project also demonstrates how such problems can be abated and reversed when SE principles and methods are applied.

References

Works Cited

INCOSE Transportation Working Group. 2011. *Systems Engineering in Transportation Projects: A Library of Case Studies*, version 1.0. Seattle, WA, USA: International Council on Systems Engineering. March 9th, 2011.

NAO. 2006. *The Modernisation of the West Coast Main Line, Report by the Comptroller and Auditor General*. London, UK: National Audit Office. November 22, 2006. HC 22 Session 2006-2007.

Railway People. 2011. '"West Coast Route Modernisation." RailwayPeople.com website. Accessed July 25, 2011. Available at: http://www.railwaypeople.com/rail-projects/west-coast-route-modernisation-3.html.

Primary References

None.

Additional References

None.

< Previous Article | Parent Article | Next Article >

SEBoK v. 1.4 released 29 June 2015

SEBoK Discussion

Please provide your comments and feedback on the SEBoK below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account. Simply type your comment in the text field below and DISQUS will guide you through the login or registration steps. Feedback will be archived and used for future updates to the SEBoK. *If you provided a comment that is no longer listed, that comment has been adjudicated. You can view adjudication for comments submitted prior to SEBoK v. 1.0 at SEBoK Review and Adjudication. Later comments are addressed and changes are summarized in the Letter from the Editor and Acknowledgements and Release History.*

If you would like to provide edits on this article, recommend new content, or make comments on the SEBoK as a whole, please see the SEBoK Sandbox [1].

ENCODED_CONTENT
MTMwMTAPGRpdiBpZD0iZGlzcXVzX3RocmVhZCI+PC9kaXY+CjxzY3JpcHQgdHlwZT0idGV4dC9qYXZhc2N
END_ENCODED_CONTENT

Singapore Water Management Vignette

This vignette describes a systems engineering approach in the development of a sustainable National Water Management System for the Republic of Singapore. It demonstrates the successful outcome of long term planning and a systems approach to preempt a critical water shortage. The vignette is primarily based on information taken from a paper presented at the INCOSE International Symposium in 2008. (Chia 2008)

Vignette Description

When Singapore achieved independence in 1965, water supply depended on water catchment in local reservoirs and two bilateral water agreements with its closest neighbor, Malaysia. These water agreements are registered with the United Nations. The first agreement expired in August 2011, and the second agreement will expire in 2061 (Singapore 2012). After several failed attempts to renegotiate the extension of the first water agreement, Singapore determined that it was necessary to achieve full water self sufficiency by 2060 in case the second water agreement also could not be extended. An intermediate goal was to match the supply of the first water agreement before it expired. This was achieved in several ways. In 2001, the Public Utilities Board (PUB), the national water agency responsible for treating raw water in Singapore, was charged to also begin managing wastewater and stormwater, allowing for an integrated and holistic approach to water management.

This vignette examines Singapore's water management system from a large-scale systems engineering perspective, particularly focusing on the goals, boundaries (see Concepts of Systems Thinking), stakeholders (see Stakeholder Needs and Requirements), and complexities involved in this type of a national system. This approach illustrates how Systems Thinking (illustrated through causal loop diagrams) and other systems engineering tools may be used to understand systems complexities. Concepts and methodologies of learning organizations were applied to enable understanding of behavioral complexities. Lean thinking facilitated a long term strategic philosophy, built on the premise of continuous improvements.

Perhaps more importantly, it shows that while systems engineering, especially the Systems Approach, is necessary for the conceptualization and planning of such a complex system, it is not sufficient for success. It is the systemic structures that have been put in place over decades, political will, leadership, people, and culture that make such tasks realizable.

The supply of water in Singapore is managed in totality. Collecting rainwater, purchasing water, purifying water utilizing reverse osmosis and desalination were all considered. Approaches included even incentivising consumers to change their habits by making drains and canals recreational areas to encourage the public not to dispose of waste in their drains. By managing sewage and drainage together with water, environmental considerations are taken into account as well. By carefully adjusting organizational boundaries, Singapore has managed to reduce silo thinking and parochial interests. The relationships between the industry innovators, government, suppliers and users, and technology innovators create opportunities for Singapore's water management. This demonstrates how multiple stakeholder interests can be combined to create a viable water management solution. Continuous improvements through the use of technology and elimination of waste, such as reducing water that is not accounted for in the system, help to assure the sustainability of an adequate supply of water for a growing Singapore population. The Singapore Water Management system is already in successful operation and is being studied by the Organisation for Economic Co-operation and Development (OECD) and by other nations.

Summary

The supply of water in Singapore is managed in totality through a systems approach, i.e., water catchment, supply, sewage and drainage. The importance of relationships between the stakeholders is also recognized. Industry innovators, political leadership, suppliers, and consumers are all involved; the project has been able to incentivize this diverse group to work together for a common goal, i.e., assuring the sustainability of an adequate supply of water for Singapore into the future.

Utilizing systems engineering and taking into consideration the systemic structures and culture required has helped Singapore achieve its first milestone of supplying its own water resources by 2010. Singapore has been able to overcome the shortfall that would have come about with the expiry of the first water agreement with Malaysia in 2011.

References

Works Cited

Chia, A. 2008. "A Large-Scale Systems Engineering Perspective of Water Management in Singapore." Proceedings of the 18th Annual INCOSE International Symposium, 15-19 June 2008, Utrech, The Netherlands.

Singapore Government. 2012. "The Singapore Water Story." Accessed August 2012. Available at http://www.pub.gov.sg/water/Pages/singaporewaterstory.aspx.

Primary References

None.

Additional References

Public Utilities Board. 2007. "PUB Main Website". Accessed August 2011. Available at http://www.pub.gov.sg.

Tortajada, C. 2006. "Water Management in Singapore." *International Journal of Water Resources Development.* 22(2): 227-240.

< Previous Article | Parent Article | Next Article >

SEBoK v. 1.4 released 29 June 2015

SEBoK Discussion

Please provide your comments and feedback on the SEBoK below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account. Simply type your comment in the text field below and DISQUS will guide you through the login or registration steps. Feedback will be archived and used for future updates to the SEBoK. *If you provided a comment that is no longer listed, that comment has been adjudicated. You can view adjudication for comments submitted prior to SEBoK v. 1.0 at SEBoK Review and Adjudication. Later comments are addressed and changes are summarized in the Letter from the Editor and Acknowledgements and Release History.*

If you would like to provide edits on this article, recommend new content, or make comments on the SEBoK as a whole, please see the SEBoK Sandbox [1].

ENCODED_CONTENT
MTEwNzEPGRpdiBpZD0iZGlzcXVzX3RocmVhZCI+PC9kaXY+CjxzY3JpcHQgdHlwZT0idGV4dC9qYXZhc2Nya
END_ENCODED_CONTENT

FAA Advanced Automation System (AAS) Vignette

In 1981 the Federal Aviation Administration (FAA) announced the Advanced Automation Program, which was established to modernize air traffic control (ATC) computer systems. A centerpiece of the project was the Advanced Automation System (AAS). AAS was the largest project in FAA's history to modernize the nation's ATC system. AAS would replace computer hardware and software as well as controller work stations at tower, terminal, and en-route facilities and allow the ATC system to accommodate forecasted large increases in traffic through the use of modern equipment and advanced software functions. (GAO 1992)

Vignette Description

The FAA originally proposed AAS in 1982 as a project that would cost $2.5 billion and be completed in 1996. However, substantial cost increases and schedule delays beset the AAS project over it history, caused by numerous problems in AAS development:

- The project began with a design competition between Hughes and IBM. The competition involved numerous extensions and took four years to complete. Analysis by the FAA and others pointed to inadequate consideration of user expectations and improper assessment of the technology risks. (Barlas 1996)
- The FAA pushed for 99.99999% reliability, which was considered by some "more stringent than on any system that has ever been implemented" and extremely costly. (DOT 1998)
- The program created unworkable software testing schedules - "Testing milestones were skipped or shortcutted and new software was developed assuming that the previously developed software had been tested and performed." (Barlas 1996)

- There were an extraordinary number of requirements changes. For example, for the Initial Sector Suite System (ISSS), a key component of AAS, there were over 500 requirements changes in 1990. Because of these changes, 150,000 lines of software code had to be rewritten at a cost of $242 million. (Boppana et al. 2006)
- IBM's cost estimation and development process tracking used inappropriate data, were performed inconsistently, and were routinely ignored by project managers. The FAA conservatively expected to pay about $500 per line of computer code - five times the industry average. The FAA ended up paying $700 to $900 per line for the AAS software. (Gibbs 1994)
- In 1988, FAA estimated that the AAS program - both contract and supporting efforts - would cost $4.8 billion. By late 1993, the FAA estimated that it would cost $5.9 billion. Before the program was dramatically restructured in 1994, estimates had risen to as much as $7 billion, with key segments expected to be behind schedule by as much 8 years. In 1994, with significant cost and schedule overruns, as well as concerns about adequate quality, usability, and reliability, the AAS program ceased to exist as originally conceived, leaving its various elements terminated, restructured, or as parts of smaller programs. (DOT 1998)

The AAS problems could be associated with the non-use or misuse of a number of systems engineering (SE) concepts and practices: system requirements, system architecture complexity, project planning, risk management, change management, system analysis and design, system reliability, system integration, system verification and system validation/testing, and management oversight.

Summary

The AAS program was the centerpiece of an ambitious effort begun in the 1980s to replace the computer hardware and software throughout the ATC system - including controller workstations, and en-route, terminal, and tower air traffic control facilities. AAS was intended to provide new automated capabilities to accommodate increases in air traffic. After sustaining serious cost and schedule problems, FAA dramatically restructured the program into more manageable pieces. This action included terminating major segments of the contract. (DOT 1998)

References

Works Cited

Barlas, S. "Anatomy of a Runaway: What Grounded the AAS." IEEE *Software*. 13(1): 104-106.

Boppana, K., S. Chow, O.L. de Weck, C. LaFon, S.D. Lekkakos, J. Lyneis, M. Rinaldi, Z. Wang, P. Wheeler, M. Zborovskiy. 2006. "Can Models Capture the Complexity of the Systems Engineering Process?" Proceedings of the International Conference on Complex Systems (ICC2006), 11-15 June 2006, Istanbul, Turkey.

DOT. 1998. *Audit Report: Advance Automation System, Federal Aviation Administration*. Washington, DC, USA: Office of Inspector General, U.S. Department of Transportation.

GAO. 1992. *Advanced Automation System Still Vulnerable to Cost and Schedule Problems*. Washington, DC, USA: United States General Accounting Office (GAO). GAO/RCED-92-264.

Gibbs, W.W. "Software's Chronic Crisis." *Scientific American*. September 1994.

Primary References

None.

Additional References

Britcher, Robert. 1999. *The Limits of Software: People, Projects, and Perspectives*. Boston: Addison Wesley.

< Previous Article | Parent Article | Next Article >

SEBoK v. 1.4 released 29 June 2015

SEBoK Discussion

Please provide your comments and feedback on the SEBoK below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account. Simply type your comment in the text field below and DISQUS will guide you through the login or registration steps. Feedback will be archived and used for future updates to the SEBoK. *If you provided a comment that is no longer listed, that comment has been adjudicated. You can view adjudication for comments submitted prior to SEBoK v. 1.0 at SEBoK Review and Adjudication. Later comments are addressed and changes are summarized in the Letter from the Editor and Acknowledgements and Release History.*

If you would like to provide edits on this article, recommend new content, or make comments on the SEBoK as a whole, please see the SEBoK Sandbox [1].

ENCODED_CONTENT
MTc3MzUPGRpdiBpZD0iZGlzcXVzX3RocmVhZCI+PC9kaXY+CjxzY3JpcHQgdHlwZT0idGV4dC9qYXZhc2Nya
END_ENCODED_CONTENT

Standard Korean Light Transit System Vignette

This vignette deals with systems engineering (SE) concepts and guidelines applied to the development of the Standard Korean Light Transit System (SKLTS). In Korea, local authorities had historically been interested in light transit to help resolve their transportation problems. The SKLTS was a joint effort between local authorities and the central government. It was built to provide a standard platform on which any local authority could construct its own light transit system. The issues of stakeholder requirements, safety, and reliability, availability, and maintainability were critical to the success of this system.

Vignette Description

The elements of the SKLTS were classified into four groups (as shown in Figure 1): trains, signal systems, electric and machinery (E&M) systems, and structures. Trains and vehicles were to be automatically operated, without need for human operators. Operation systems and their interfaces were based on digital signals and communications. For SKLTS, SE-based design activities focused on reliability, availability, maintainability, and safety (RAMS), and were integrated into project management (PM) activities during all phases.

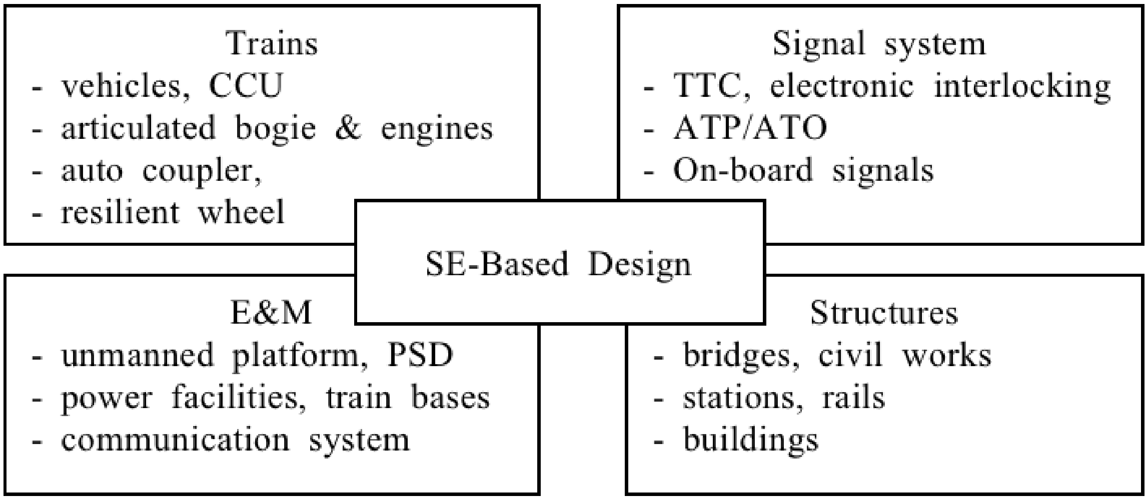

Figure 1. Subsystems of the SKLTS (Ahn, 2005). (Notes: CCU: Central Control Unit; TTC: Total Traffic Control; ATP: Automatic Train Protection; ATO: Automatic Train Operation; PSD: Platform Screen Door) Reprinted with permission of *Journal of the Korean Society for Railway*. All other rights are reserved by the copyright owner.

The project life cycle for the SKLTS is summarized in Figure 2. It consisted of 7 phases: concept studies, concept development, preliminary design, design, system production and testing, performance evaluation, and operation/maintenance/close-out (OMC) - please see (Choi 2007) and (Chung et al. 2010) for further details. These phases, with the exception of the production and test phases, are completed through an evaluation and decision point (EDP) (milestone), depicted as a colored circle in Figure 2. These EDPs correspond to common life cycle artifacts such as requests for proposal (RFPs), proposals, preliminary design reviews (PDRs), and critical design reviews (CDRs).

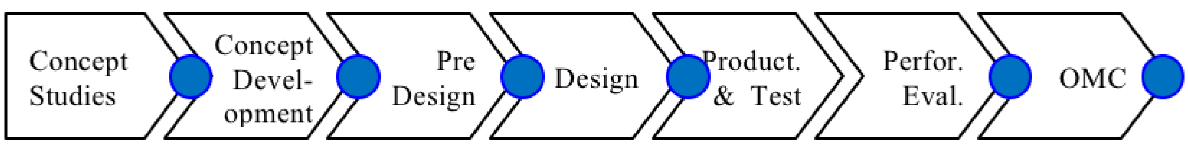

Figure 2. 7 phases of the SKLTS development (Ahn 2005). Reprinted with permission of the *Journal of the Korean Society for Railway*. All other rights are reserved by the copyright owner.

During the SKLTS development, SE activities were focused on RAMS as summarized in Table 1.

Table 1. The SE Framework of the SKLTS (Ahn 2005). Reprinted with permission of the *Journal of the Korean Society for Railway*. All other rights are reserved by the copyright owner.

Phases	Safety	Reliability	Function	Performance
Concept studies	• Requirements analysis	• Identifying RAM conditions • RAM allocation	• System configuration • Interface management	• Performance simulation
Concept development & pre-design	• Safety planning • Defining safety procedures & levels	• RAM planning • Initial availability analysis	• Defining scenarios and alarm procedure • Pre-designing command rooms	• Interface analysis

Design	• Hazard log • Safety case analysis • Risk analysis	• Reporting RAM analysis • RAM analysis of auxiliary systems	• Defining alarm systems • Train analysis • Functionality analysis of stations	• Interface analysis
Performance evaluation	• Safety test planning & testing	• Verification and Validation (V&V) RAM • Maintainability test	• System test planning and testing	• Performance test planning & testing
Initial Operation	• System acceptance • Driver certification	• RAM monitoring • FRACAS*	• Analyzing systems • Identifying improvement points	• Performance monitoring

*FRACAS: Failure Reporting & Corrective Action System

In the "concept studies" and "concept development" phases, requirements included the RAMS objectives. Planning activities in this phase included the scheduling of various tests and evaluations to be conducted after system design. The basic layout of rails and command rooms was also proposed. Finally, it was during this phase that interface management procedures and relationships between requirements and systems were defined. For RAMS engineering, it was also important to establish associated plans and criteria (e.g., RAM plans, safety plans, service availability, etc.).

During the pre-design phase, the basic architecture of the system was determined for safety planning, RAMS planning, and operational scenarios. Interfaces among subsystems were defined as well as management procedures for contractors and legal regulations. The functional analysis dealt with timeline, accuracy of stop points, and trip times. Pre-design activities also included the specifications of major system elements such as signal systems, trains, and interfaces. For RAMS engineering, safety scenarios were defined, and the hazard and risk analyses were performed.

During the design and performance evaluation phases, hazard log and RAMS analyses were performed to ensure that each subsystem met safety requirements. The specifications of alarm systems and stations were also defined. In addition, V&V and test procedures were determined for performance evaluation. During the design phase, a design/construction interface manual (D/CIM) was developed and applied to ensure integrated and consistent design. (Bombardier, 2005)

Because SKLTS was designed as an automatically-driven system, RAMS issues were critical to its success. The safety and reliability of the SKLTS were evaluated on a test railway that was constructed to standard specifications. Data was gathered from existing Korean light rail systems, as well as the light rail systems from other countries, to support V&V activities.

Various methods were applied for achieving the RAMS objectives, including RAMS requirements analysis, safety and RAMS planning, utilization of systems scenarios, and construction risk analysis.

Initial operation of SKLTS was allowed only after the system was formally accepted and operators were properly certified. During test operation, RAMS performance was continuously monitored and system scenarios were used successfully to evaluate the dynamic behavior of the system. A failure reporting and corrective action system (FRACAS) was used to gather accident and failure data. Continuous improvement when the system is in normal operation was identified as a requirement; the results from the FRACAS will be used to support improvement of the system, maintenance, and improvement of procedures.

Summary

Korean local authorities have successfully introduced the SKLTS to their precincts with some modifications. Successful examples include the Inchun Airport Line and the Seoul 9th Subway Line. One lesson learned identified was that requirement analysis, especially in the first few phases, should have been more complete.

References

Works Cited

Ahn, S.H. 2005. “Systems Engineering Applied to the Construction of Unmanned Light Rail Transit Systems.” *Journal of the Korean Society for Railway.* 8(2): 41-49.

Bombardier. 2005. *Design/Construction Interface Manual*. Montréal, Québec, Canada: Bombardier.

Choi, Y.C. 2007. "Systems Engineering Application Model for the National R&D Project: Focusing on the Railway Systems." Ph.D. dissertation, Ajou University, 2007.

Chung, S. Y., S.G. Lee, D.W. Lee, and S.T. Jeon. 2010. “A Study on The Application Model of Systems Engineering to Advance the Business of the Light Rail Transit (LRT).” Proceedings on the Autumn Conference of the Korean Society for Railway, p. 24-30.

Primary References

None.

Additional References

Han, S.Y. and A.H. Lee. 2005. *System Engineering for The Technology Development Program of Light Transit Systems: Internal Research Report.* Gyeongi-do, Korea: Korea Railroad Research Institute.

Korean Agency for Technology and Standards. 2009. *KSX ISO/IEC 15288: Life Cycle Processes of IT Systems Based on Systems and Software Engineering, Research Report.* Gyeonggi-do, Korea: Korean Agency for Technology and Standards.

Lee, W.D. 2002. “A Study on the Top-Level Functional Analysis for the Automated Guideway Transit by System Engineering Tools.” Proceedings of the Autumn Conference of the Korean Society for Railway, p. 202-207.

< Previous Article | Parent Article | Last Article (Return to TOC)

SEBoK v. 1.4 released 29 June 2015

SEBoK Discussion

Please provide your comments and feedback on the SEBoK below. You will need to log in to DISQUS using an existing account (e.g. Yahoo, Google, Facebook, Twitter, etc.) or create a DISQUS account. Simply type your comment in the text field below and DISQUS will guide you through the login or registration steps. Feedback will be archived and used for future updates to the SEBoK. *If you provided a comment that is no longer listed, that comment has been adjudicated. You can view adjudication for comments submitted prior to SEBoK v. 1.0 at SEBoK Review and Adjudication. Later comments are addressed and changes are summarized in the Letter from the Editor and Acknowledgements and Release History.*

If you would like to provide edits on this article, recommend new content, or make comments on the SEBoK as a whole, please see the SEBoK Sandbox [1].

ENCODED_CONTENT
MTgyMTQPGRpdiBpZD0iZGlzcXVzX3RocmVhZCI+PC9kaXY+CjxzY3JpcHQgdHlwZT0idGV4dC9qYXZhc2Nya

END_ENCODED_CONTENT

Article Sources and Contributors

Systems Engineering Implementation Examples *Source*: http://sebokwiki.org/d/index.php?oldid=50215 *Contributors*: Bkcase, Dhenry, Dholwell, Hdavidz, Jgercken, Kguillemette, Rturner, Skmackin, Smenck2, Wikiexpert, Zamoses

Matrix of Implementation Examples *Source*: http://sebokwiki.org/d/index.php?oldid=51387 *Contributors*: Alee, Apyster, Asquires, Bkcase, Dhenry, Dholwell, Hdavidz, Jbrackett, Jgercken, Kguillemette, Mhenshaw, Radcock, Rturner, Skmackin, Smenck2, Thilburn, Wikiexpert, Zamoses

Case Studies *Source*: http://sebokwiki.org/d/index.php?oldid=51383 *Contributors*: Alee, Apyster, Bkcase, Bwhite, Cnielsen, Dcarey, Dhenry, Dholwell, Hdavidz, Jgercken, Kguillemette, Radcock, Skmackin, Smenck2, Wikiexpert, Zamoses

Successful Business Transformation within a Russian Information Technology Company *Source*: http://sebokwiki.org/d/index.php?oldid=50318 *Contributors*: Bkcase, Bwhite, Cnielsen, Kguillemette, Radcock

Federal Aviation Administration Next Generation Air Transportation System *Source*: http://sebokwiki.org/d/index.php?oldid=50315 *Contributors*: Bkcase, Bwhite, Cnielsen, Kguillemette, Radcock

How Lack of Information Sharing Jeopardized the NASA/ESA Cassini/Huygens Mission to Saturn *Source*: http://sebokwiki.org/d/index.php?oldid=50227 *Contributors*: Bkcase, Bwhite, Kguillemette, Radcock

Hubble Space Telescope Case Study *Source*: http://sebokwiki.org/d/index.php?oldid=51075 *Contributors*: Apyster, Asquires, Bkcase, Bwhite, Dhenry, Dholwell, Hdavidz, Jgercken, Radcock, Rfreeman, Smenck2, Wikiexpert, Zamoses

Global Positioning System Case Study *Source*: http://sebokwiki.org/d/index.php?oldid=51120 *Contributors*: Apyster, Asquires, Bkcase, Bwhite, Dhenry, Dholwell, Hdavidz, Jgercken, Radcock, Rfreeman, Smenck2, Thilburn, Wikiexpert, Zamoses

Global Positioning System Case Study II *Source*: http://sebokwiki.org/d/index.php?oldid=51119 *Contributors*: Bkcase, Bwhite, Radcock

Medical Radiation Case Study *Source*: http://sebokwiki.org/d/index.php?oldid=50969 *Contributors*: Apyster, Bkcase, Dhenry, Dholwell, Hdavidz, Jbrackett, Jgercken, Kguillemette, Smenck2, Wikiexpert, Zamoses

FBI Virtual Case File System Case Study *Source*: http://sebokwiki.org/d/index.php?oldid=46934 *Contributors*: Apyster, Bkcase, Dhenry, Dholwell, Hdavidz, Jbrackett, Jgercken, Skmackin, Smenck2, Wikiexpert, Zamoses

MSTI Case Study *Source*: http://sebokwiki.org/d/index.php?oldid=51393 *Contributors*: Apyster, Bkcase, Dhenry, Dholwell, Hdavidz, Jgercken, Skmackin, Smenck2, Wikiexpert, Zamoses

Next Generation Medical Infusion Pump Case Study *Source*: http://sebokwiki.org/d/index.php?oldid=50573 *Contributors*: Apyster, Bkcase, Dhenry, Dholwell, Hdavidz, Jgercken, Smenck2, Wikiexpert

Design for Maintainability *Source*: http://sebokwiki.org/d/index.php?oldid=51378 *Contributors*: Bkcase, Radcock

Complex Adaptive Operating System Case Study *Source*: http://sebokwiki.org/d/index.php?oldid=51379 *Contributors*: Bkcase, Radcock

Vignettes *Source*: http://sebokwiki.org/d/index.php?oldid=50963 *Contributors*: Bkcase, Dhenry, Dholwell, Hdavidz, Jgercken, Skmackin, Smenck2, Thilburn, Wikiexpert, Zamoses

Denver Airport Baggage Handling System Vignette *Source*: http://sebokwiki.org/d/index.php?oldid=51389 *Contributors*: Apyster, Asquires, Bkcase, Dhenry, Dholwell, Hdavidz, Jgercken, Kguillemette, Qwang, Skmackin, Smenck2, Thilburn, Wikiexpert, Zamoses

Virginia Class Submarine Vignette *Source*: http://sebokwiki.org/d/index.php?oldid=46755 *Contributors*: Apyster, Asquires, Bkcase, Dhenry, Dholwell, Hdavidz, Jgercken, Skmackin, Smenck2, Wikiexpert, Zamoses

UK West Coast Route Modernisation Project Vignette *Source*: http://sebokwiki.org/d/index.php?oldid=46740 *Contributors*: Alee, Apyster, Asquires, Bkcase, Dhenry, Dholwell, Hdavidz, Jgercken, Rturner, Skmackin, Smenck2, Thilburn, Wikiexpert, Zamoses

Singapore Water Management Vignette *Source*: http://sebokwiki.org/d/index.php?oldid=46710 *Contributors*: Alee, Apyster, Asquires, Bkcase, Dhenry, Dholwell, Hdavidz, Jgercken, Rturner, Skmackin, Wikiexpert, Zamoses

FAA Advanced Automation System (AAS) Vignette *Source*: http://sebokwiki.org/d/index.php?oldid=51390 *Contributors*: Apyster, Asquires, Bkcase, Dhenry, Dholwell, Hdavidz, Jgercken, Rturner, Thilburn, Wikiexpert

Standard Korean Light Transit System Vignette *Source*: http://sebokwiki.org/d/index.php?oldid=47985 *Contributors*: Apyster, Asquires, Bkcase, Ccalvano, Dhenry, Dholwell, Hdavidz, Jgercken, Mhenshaw, Rturner, Skmackin, Smenck2, Wikiexpert, Zamoses

Image Sources, Licenses and Contributors

File:MissionAndCapabilities.png *Source*: http://sebokwiki.org/d/index.php?title=File:MissionAndCapabilities.png *License*: unknown *Contributors*: Bkcase
File:V Model Process of Transformation.png *Source*: http://sebokwiki.org/d/index.php?title=File:V_Model_Process_of_Transformation.png *License*: unknown *Contributors*: Bkcase
File:Results of Tranformation.png *Source*: http://sebokwiki.org/d/index.php?title=File:Results_of_Tranformation.png *License*: unknown *Contributors*: Bkcase
File:ChoeKimFigure1.png *Source*: http://sebokwiki.org/d/index.php?title=File:ChoeKimFigure1.png *License*: unknown *Contributors*: Smenck2, Smurawski
File:ChoeKimFigure2.png *Source*: http://sebokwiki.org/d/index.php?title=File:ChoeKimFigure2.png *License*: unknown *Contributors*: Smenck2, Smurawski