

Exploring Adaptability Through Change Cases

J.D. Baker

Principal Technologist, K-LASSE, Inc and
Principal Systems Engineer, APG

Armstrong Process Group, Inc.
www.aprocessgroup.com

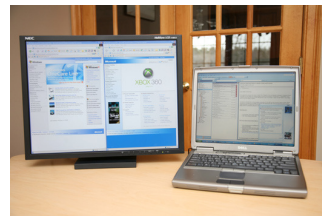
Copyright © 2010, Armstrong Process Group, Inc., All rights reserved



Change Cases

2

- A mechanism for evaluating change requests or exploring the change space of a system
- Supported by use cases and evaluation scenarios
- Intended to be applied through-out the development process



Copyright © 2010, Armstrong Process Group, Inc., All rights reserved



Change Case vs. ATAM Exploratory Scenario ³

- ATAM Growth and Exploratory Scenario
 - Stress an almost finished architecture
 - Formally developed
- Evaluation of Change Cases
 - Applied at any point in the development
 - Semi-formal

Copyright © 2010, Armstrong Process Group, Inc., All rights reserved

APG

Categories of Change Cases - HW ⁴

- Ports to new platforms or new hardware configurations
 - Changes to peripherals
 - Changes to how things are connected

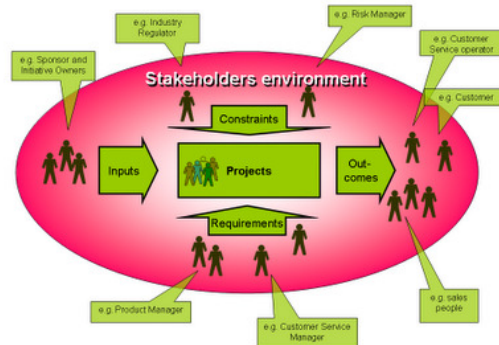


Copyright © 2010, Armstrong Process Group, Inc., All rights reserved

APG

Categories of Change Cases - Stakeholders

- New Stakeholders/Actors to support
- Evolving/changing interests and focus of key stakeholders



<http://www.betterprojects.net/2007/05/introduction-to-stakeholder-management.html>

Copyright © 2010, Armstrong Process Group, Inc., All rights reserved

APG

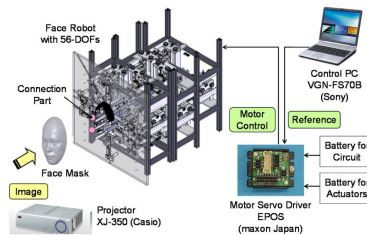
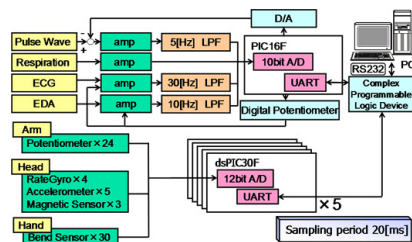
Categories of Change Cases - Functionality

- New Configurations of existing systems
- New functionality added to existing systems

AUDIO/VIDEO, CAMERAS, COMPUTERS

Filmmaker Wants to Replace False Eye With Webcam

BY DARREN MURPH (RSS FEED) — DEC 5TH 2008 AT 12:16PM



Copyright © 2010, Armstrong Process Group, Inc., All rights reserved

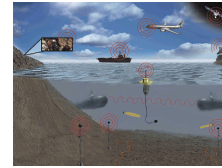
APG

Categories of Change Cases – Domain Changes

- Application domain changes and new domains
 - Temperature
 - Air pressure
 - Usage – unintended use

[How To Use Your TV As a Computer Monitor](#)

Complete How To Guide On This Topic and 1000s of Others subscribe now.
[how-to-do-anything.net](#)



Copyright © 2010, Armstrong Process Group, Inc., All rights reserved

APG

Content

- Content of the change case used in the evaluation of a system architecture or design
 - Name – one sentence describing the change
 - Identifier – unique identifier for this change case, e.g. CC17
 - Description – sufficient detail to allow evaluation; exploratory change cases require less detail
 - Likelihood – rating of how likely change case is to occur (e.g. High, Medium, Low or %)
 - Timeframe – indication of when the change is likely to occur
 - Potential Impact – indication of how drastic the impact will be if the change occurs
 - Source – indication of where the requirement came from
 - Notes

Copyright © 2010, Armstrong Process Group, Inc., All rights reserved

APG

Evaluating the Technique ⁹

- Structured format for evaluating the impact of a change case?
 - One possibility is a variation on Robustness Analysis. This technique comes from some of Ivar Jacobson's early work, as elaborated by Doug Rosenberg in the ICONIX process.
- Emulate the analysis of growth scenarios in the SEI's Architecture Trade-off Analysis Method (ATAM).
 - Will the development team have sufficient expertise to make this useful?

Copyright © 2010, Armstrong Process Group, Inc., All rights reserved



Evaluating the Technique (2) ¹⁰

- Thus far only the notion of use case scenario walk-throughs has been employed
 - Still looking for a situation where I can apply the Robustness Analysis variant
- Much of the initial effort focused on applying change cases to evaluating requirements changes
 - New features for an existing product were explored, with a constraint that no existing features could be removed.
 - Requirements were then passed to the systems architecture team

Copyright © 2010, Armstrong Process Group, Inc., All rights reserved



Developing the Change Cases

11

- **Name** – one sentence describing the change
 - this took more time to get right than I had anticipated
 - Important to get right since it helped focus the evaluation
 - Make sure the team understands exactly what kind of change is expected
- **Identifier** – unique identifier for this change case, e.g. CC17
 - As more change cases are added, it helps to have a unique identifier for them

Copyright © 2010, Armstrong Process Group, Inc., All rights reserved



Developing the Change Cases (2)

12

- **Description** – sufficient detail to allow evaluation; exploratory change cases require less detail
 - Iterative development of change cases is possible and useful
 - The required amount of detail seems to be project specific but more work needs to be done to be certain
- **Likelihood** – rating of how likely the change case is to occur (e.g. High, Medium, Low or %)
 - Best to do this assessment early in the development
 - The name as a sentence was enough for us to assign a value using an ATAM-like voting scheme
 - Focused on the High-High changes

Copyright © 2010, Armstrong Process Group, Inc., All rights reserved



Developing the Change Cases (3)

13

- **Timeframe** – indication of when the change is likely to occur
 - This was addressed as the third step in our process, following Name and Likelihood
 - Focused our efforts on the things happening soonest
- **Potential Impact** – indication of how drastic the impact will be if the change occurs
 - This turned out to be the most challenging to element to complete

Copyright © 2010, Armstrong Process Group, Inc., All rights reserved



Developing the Change Cases (4)

14

- **Source** – indication of where the requirement came from
 - Not all stakeholders are created equal
 - We learned quickly where the major source of changes was
- **Notes**
 - It seems like we always had something extra to say.

Copyright © 2010, Armstrong Process Group, Inc., All rights reserved



Summary

15

- Change Cases are a reasonable way to evaluate architecture requirements
- The systems engineering team learned the technique quickly and was able to make useful decisions
 - With the caveat that it was only one team
- More work needs to be done

Copyright © 2010, Armstrong Process Group, Inc., All rights reserved



Background

16

- Change Case Analysis
- A while back, another architect mentioned the topic of change cases to me when we were discussing ways to evaluate software architectures. That was the first time I had ever heard the term, and I assumed it was a new technique. I filed it away as a topic to research later. Recently I was asked to provide some training on the topic of architecture evaluation and I decided to do that postponed research.
- The two years I spent as a Guide at About.com taught me a lot about internet searches, so I was surprised when I was unable to turn up much in the way of relevant material (if you need to change text from upper to lower case or vice versa, I can tell you where to look). The only truly relevant information was on Scott Ambler's Agile Modeling site - <http://www.agilemodeling.com/artifacts/changeCase.htm>. The most significant thing there was a link to the book Designing Hard Software by Douglas W. Bennett (<http://www.amazon.com/exec/obidos/ASIN/0133046192>), an out of print 1997 book. The only reasonable thing for me to do was buy one of the used books available on Amazon.
- Bennett does a good job of identifying categories of change cases. Even if some of the specific examples seem a little out-dated, the concepts translate quite readily into problems being faced by systems under development today. What's missing is any indication of what the content of the change case should be, and how they should then be used in the evaluation of a software system architecture and design.

Copyright © 2010, Armstrong Process Group, Inc., All rights reserved



Background (2)

17

- Scott Ambler offers the following as a change case template:
 - Name: (One sentence describing the change)
 - Identifier (A unique identifier for this change case, e.g. CC17)
 - Description (A couple of sentences or a paragraph describing the basic idea)
 - Likelihood (Rating of how likely the change case is to occur (e.g. High, Medium, Low or %))
 - Timeframe (Indication of when the change is likely to occur)
 - Potential Impact (Indication of how drastic the impact will be if the change occurs)
 - Source (Indication of where the requirement came from)
 - Notes

Copyright © 2010, Armstrong Process Group, Inc., All rights reserved



Background (3)

18

- That is a start on an answer to the first question - what is the content of a change case? But that content list raises a question of its own - how do we decide on the potential impact? Also, we still need to consider how we use this information. Is there a structured format for evaluating the impact of a change case?
- One possibility is a variation on Robustness Analysis. This technique comes from some of Ivar Jacobson's early work, as elaborated by Doug Rosenberg in the ICONIX process. In addition to the books that include this topic (<http://iconixprocess.com/books/>) there's an article on the ICONIX web site (<http://iconixprocess.com/iconix-process/analysis-and-preliminary-design/robustness-analysis/>) and one on the Dr. Dobb's web site (<http://www.ddj.com/architect/184414712>). Robustness Analysis is applied to use cases in order to verify their completeness. A variation might help to identify what objects/components will need to have additional responsibilities and what new components might need to be added where it doesn't make sense to change or extend an existing component.
- A less structured approach might be to emulate the analysis of growth scenarios in the SEI's Architecture Trade-off Analysis Method (ATAM). SEI evaluators often talk about pulling a thread to drill down wherever it is necessary to determine whether or not the architecture will support the growth. This approach works well for the experienced evaluators at the SEI, but may be quite challenging for someone without their level of experience.
- My plan is to experiment with these techniques. If you have any suggestions for an alternative method, or a way to apply these two techniques, then please feel free to comment.

Copyright © 2010, Armstrong Process Group, Inc., All rights reserved

