Software development at NAV Portugal

ES2 WS3-11 Bled/Slovenia 21/22 September 2011
Software Safety Assurance & Degraded Modes of Operations

Paula Santos
Once upon a time
Software development at NAV Portugal

• Pros & Cons of internal developments
• The process
  • Difficulties
  • Achievements
• Regulation
Pros & Cons

- We all belong to the same company
- We are a team
- We live in the same building
- Informal chains are cherished
- High commitment
Pros & Cons

- Proximity
  - Every day learning
  - Speaking the same dialect
    - Mutual understanding
  - Informal communication channels
    - Earlier knowledge of “news”
    - Explore different approaches
    - Access other lines of thinking
  - Insight knowledge
    - Allows counter proposals
- Short change chain
  - Product aligned with “dream”
Pros & Cons

• Easy changing
  • Too many change requests
  • Immature change requests
  • Request for changes too late (It’s so easy…)

• Blurred frontiers (Operational / Technical staff)
  • Who takes the lead?
  • Who does what?
  • Project team
    • Availability / in competition with other work
Process

**Ops involved**
- Operational specifications
- FHA report
- System specification
- Acceptance tests

**Tech only**
- System architecture
- Is architecture safe?
- Integration tests
- Component tests
- Software requirements
- Code ...
- Unit tests

Software requirements: Code ...
Integration tests: Component tests
Is architecture safe?: Software requirements
Acceptance tests: Integration tests
FHA report: System specification
Operational specifications: Code ...
The Process

- Project Planning and Control
  - System Analysis
  - System Development
  - System Delivery
  - System Testing & Evaluation

- Global
  - Quality Management
  - Problem reports Management
  - Change requests Management

- Operational environment
  - Intervention on operational systems
  - Operational configuration management

- Development environment
  - Development configuration Management
  - Test platform management
Produced documents

- Operational specifications
- FHA
- Regulation

System delivery
- Delivery Document
- Conformity Declaration
- Technical File
- Transition Plan

Project Planning & Control
- PMP
- Status reports
- Delivery Document

System Analysis
- System Specifications
- Operator Handbook
- External interfaces
- System Architecture
- FHA
  - Is the architecture safe?

System Testing & Evaluation
- Test Management plan
- Test descriptions
- Integration Test reports
- Change Proposals
- Problem Reports
- Installation Manual
- Operator Handbook

System Development
- SW requirements
- SW design
- Internal interfaces
- Unit Test reports
- Subsystem test reports
- Installation Manual
• What’s in a standard?

IEC 12207_2008: (Purpose, Outcomes, Activities and tasks)

“As a result of a successful implementation of System Requirements Analysis:
   a) A defined set of system functional and non-functional requirements describing the problem to be solved are established
   (...)”

• Help, how?
  – Building instructions
  – Examples
  – Reviews
### How to - Building instructions

- **Use Case** (with the Use Case name, unique Use Case identification, reference to the corresponding operational specification)
- **Brief Description** - brief description of the Use Case meaning
- **Preconditions** - Previous defined conditions that must satisfy the beginning of the Use Case
- **Flow of Events** - a section for the basic path and each alternative path
- **Post Conditions** - list of conditions that must be true when the use case ends successfully
- **Priority** - choose one of the following alternatives:
  - Not defined
  - Low priority
  - Medium priority
  - High priority
- **Source** - where the requirement comes from, who asked for it? For example: ICAO document; meeting xxx, stakeholder xxx.

<table>
<thead>
<tr>
<th><strong>Use Case</strong></th>
<th>Change FPL Field</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ID - Use Case</strong></td>
<td>XXXX-DES-Change_FPL</td>
</tr>
<tr>
<td><strong>OS Ref</strong></td>
<td>XXXX_EO_UC_CHG_FPL3</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>Field change on a flight plan</td>
</tr>
<tr>
<td><strong>Preconditions</strong></td>
<td>The FPL window is open and contains flight plans</td>
</tr>
</tbody>
</table>

**Happy Path**

The Use Case begins when the user marks a register and selects the Edition option;
1. The system opens a dialog window (see window ID XXX definition);
2. For each field to change
   a. The user selects the field he wishes to change;
   b. The user changes the information;
   c. The system verifies the data inserted, when the focus gets out of the changed field;
3. The user selects the OK button in order to end the change data introduction;
4. The system validates the information, the window is closed and returns to the initial screen.

<table>
<thead>
<tr>
<th><strong>Post Conditions</strong></th>
<th>The data is stored in the database.</th>
</tr>
</thead>
</table>

**Alternative Paths**

**A - Cancel**
At any point the user selects the Cancel option
1. The system ignores all the data inserted and returns to the initial values, without any data change.
2. The system closes the dialog window.

<table>
<thead>
<tr>
<th><strong>Priority</strong></th>
<th>High priority</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Source</strong></td>
<td>Project meeting on xx.mm.dd, ICAO requirement.</td>
</tr>
</tbody>
</table>
Code Reviews / walkthroughs

- Are systematically done
- Are registered (Who, when, ...)
- Allow sharing of experience
- Improve product quality
• What is the document for?
  – For auditors?
  – To say it’s done?
  – To archive?
  – For someone to use?

Doing for the sake of doing them causes frustration and bad documents – crap.
• What are readers looking for?

- I would like to understand how the system works
- What SW modules can cause “Radar picture corruption”?
- I’m trying to know what the XYZ function does…
• Planning & Management (PMP, TMP)
• Tests
• Requirements
  – System (High level) - Software
  – Funcional – Non funcional
  – Traceability
• Architecture
• Development
• Go back ...

The whole team followed the FHA (2004) and SAF-SW (2006) courses
• Regulation cocktail
  – EC 552/2004
    • Declaration of Conformity or suitability for use of constituents
    • Declaration of Verification of Systems
    • Essential requirements – ER3 - Safety
  – EC 2096/2005
    • Annex I, 3 – Safety and Quality Management
    • Annex II, 3 – Safety Management System
  – EC 1315/2007
    • Safety oversight
  – EC 482/2008
    • Software Safety Assurance
  – EC 1070/2009 (amending regulations...)
  – Implementing Rules
• Inform all planned changes on operational systems for the year
• Do safety assessment
• Inform about outcome of safety assessment (*Develop change*)
• Before going into operation, send
  – Declaration (DoC of DSU)
  – Declaration of verification of systems
  – Technical file
    • Almost all project documentation
    • Wait for questions
• How to avoid having “regulator focused documents”, i.e. Documents made just to please the regulator?

• How can one give a satisfactory answer to all the regulation requirements using:
  – The results of the safety analysis integrated with
  – The outputs of a good software engineering practice