Cockpit – A Unified Model for Product Development

- In an online world, requirements should not be isolated.
- VOCs, Hazards, FMEAs, mitigation plans, tests, protocols, critical parameters, action items, costs, etc., everything can be unified.
Bottom Line

- Selecting a Requirements Management tool is tricky
  - You have many choices
    - Some are safe
      - Being the oldest, most widely installed
    - Some are client/server
    - Some are web-based
    - Some are tuned for software development
    - Some are tuned for additional development
    - Some have module pricing and some a single price

- Whichever you choose, your users will tell you if you got it right
  - They will either want to use it or they will avoid it at all costs…
  - If they want to use it, you did well…
  - If not, you should try again
Introduction:

- Meets and exceeds world wide Requirements Management Standards:
  - Configuration Management
  - Verification and Validation
  - System Modeling
  - Document Standards
  - Import and Export

- Users will want to use it
  - One intuitive web interface for all functions
  - Easy to learn and get stuff done
  - Guided templates for deliverables and process steps
  - Automatic output of deliverables and generation of trace matrices

- Single system where everything is connected in a Unified Model
  - Requirements Management
  - Test Management
  - VOC/Needs Management
  - Risk Management
  - Critical Parameter Management
  - Cost Management
  - Meeting and Action Item Management
Requirements Management with Cockpit

Like most other Requirements Management tools on the market, Cockpit allows you to import requirements from multiple external sources such as Excel, Word, XML, and other competitive tools.
Requirements Management with Cockpit

- Typical Requirements Management document in Cockpit
  - The documents are dynamic pages with content produced from data in the database.
- They support requirement creation, editing, reviewing, approving, markups, etc.
- This format is tabular in style, but you can have any format you like.
Requirements Management with Cockpit

- At the top of each document, there is an Export command. When clicked, Cockpit produces a Word document, identical to the “on-line” version. Headers, footers, table of contents, images, tables, etc. are all supported properly.
Requirements Management with Cockpit

- Privileging in Cockpit is fully supported.

- Pick any item and select the “Access Rights.” A page will appear allowing you to set view, modify, delete and other privileges.
Requirements Management with Cockpit

- Configuration management is supported in Cockpit.

- This is true at all levels: item, document, folder, and project levels.

- The state transition is shown to the right, but basically, versions are baselined and revisioned as needed and each time an Approval process may be invoked.
Requirements Management with Cockpit

- Visual differencing is also supported.

- Clicking on the “Diff” command at the top of the page will open a new window with redlines for deleted text, green for newly added text, and so on.

- All versions of a document may be compared in this way.
Requirements Management with Cockpit

- From a higher perspective, such as the Project level, differences can also be tracked.

- To start, a project is baselined, and thereafter, multiple working versions can be created and further baselined.

- A complete audit of new, deleted, and modified items is retrieved from the project’s history page.
Requirements Management with Cockpit

- Flowing down top level requirements into lower level subsystem/component requirements is completely supported.

- Trace matrices are automatically generated and change (stability) notices are propagated through the requirement hierarchy.
Requirements Management with Cockpit

- Whatever you are trying to design in Cockpit, you can create a model for it.
- In this case, Cockpit is being used to create a use-case model.
- Additionally, Cockpit has a bi-direction link with MS Visio.
Requirements Management with Cockpit

- The Requirements Management functionality in Cockpit is directly connected to its Verification and Validation functionality.

- The Test Plan, for example, is very efficient to create, because templates allow requirements written earlier (in the PRD) to be automatically included.
Requirements Management with Cockpit

- Can allocate requirements to tests.
- Cockpit captures many-to-many relationships, either through clicking “allocation matrices” or through many other supported means in Cockpit.
Requirements Management with Cockpit

- One of Cockpit’s greatest strengths is traceability throughout the process.
- Getting a tabular trace from requirements to tests or vice versa is a simple mouse click.
Requirements Management with Cockpit

- Writing Test Protocols in Cockpit is also supported.
- Templates can be tweaked to comply with your format and any number of sections can be defined, such as setup, acceptance criteria, execution steps, results, etc.
Requirements Management with Cockpit

- Cockpit will “auto-populate” the acceptance criteria section for you.
- How? Earlier, you linked tests and requirements together—and Cockpit can use those links to fill out this section.
- If there are changes to the requirements, they will be reflected in this document as well.
Requirements Management with Cockpit

- When you write test methods, Cockpit keeps track and is building libraries of them.

- They’re reusable and can be incorporated in any number of documents.
Requirements Management with Cockpit

- Cockpit allows you to define multiple execution “milestones” for the tests in any protocol, for example “Prior to Clinical”, “Prior to Submission”, etc.

- All the text you write can be general or milestone specific and you can reuse/toggle the document to correspond to the test milestone you are performing.
Requirements Management with Cockpit

- Cockpit captures results from each test execution.

- There are no restrictions; if a test will verify more than one requirement, the results for each are captured and presented clearly.

- You can choose any format you like, and the one above shows an action item column making it easy to track issues and activities.
Requirements Management with Cockpit

- Efficiency is important.

- If you’re having a meeting and you want to have a larger review of the test results, you can toggle the format to show all results from each milestone, as well as action items, notes, etc.
To support quick decisions and review, Cockpit has many built-in reports showing the current status of items: for example, test coverage, number of failures, etc.

These reports can be run from any level, such as document, folders, or project level.
Requirements Management with Cockpit

- So, how do Cockpit’s baseline capabilities compare to other Requirements Management systems?
Baseline Requirements Management Features

- We believe you should review each of your Requirements Management tool choices. INCOSE (International Council on Systems Engineering) has a reference available comparing many vendors. The link below will take you to the full report, and we’ve included a subset of that report in the next few slides.
Baseline Requirement Management Features

- We believe you should review each of your Requirements Management tool choices.

- INCOSE (International Council on Systems Engineering) has a reference available comparing many vendors.

- The image link below will take you to the full report, and we’ve included a subset of that report in the next few slides.

NOTE: AS OF 8/16/2013 THE INCOSE TOOLS DATABASE IS DOWN FOR MAINTENANCE. CHECK BACK SOON.
## INCOSE Requirements Management Tools Survey

### INCOSE | TDWG Home | Tools Database: RM SA Measurement General

<table>
<thead>
<tr>
<th>Tools</th>
<th>IBM Rational RequisitePro (Updated 10 Oct 09)</th>
<th>SLATE Require 5.5 (A UGS Teamcenter Product)</th>
<th>UGS Teamcenter 2005</th>
<th>Coordination Context</th>
<th>Telelogic DOORS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Capturing Requirements</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>1.1 Input document enrichment/analysis</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>1.1.1 Input document change/comparison analysis</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>1.2 Automatic parsing of requirements</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>1.3 Interactive/semi-automatic requirement identification</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>1.4 Manual requirement identification</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>1.5 Batch-mode operation</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>1.6 Requirement classification</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>2. Capture System Element structure (if so, how? As document paragraphs? product structures?...)</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>2.1 Graphically capture systems structure</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>2.2 Textually capture of system structure</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>3. Requirements Flowdown</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>3.1 Requirements derivation (req. to req., req. to analysis/fitted)</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>3.2 Allocation of performance requirements to system elements (weight, risk, cost, etc.)</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>3.3 Bi-directional requirement linking to system elements</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>3.4 Capture of allocation rationale, accountability, test/verification, criticality, issues, etc. If so, how and what mechanism does it use</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
</tbody>
</table>
## INCOSE Requirements Management Tools Survey

![INCOSE Logo](image)

**INCOSE | TDWG Home | Tools Database: RM SA Measurement General**

<table>
<thead>
<tr>
<th>Tools:</th>
<th>IBM Rational RequisitePro (updated 10 Oct 06)</th>
<th>SLATE REquire 6.5 (A UGS Teamcenter Product)</th>
<th>UGS Teamcenter 2005 (Requirements Management)</th>
<th>Cognition Cockpit 5</th>
<th>Telelogic DOORS</th>
</tr>
</thead>
<tbody>
<tr>
<td>7. Groupware</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.1 Support of concurrent review, markup, &amp; comment</td>
<td><strong>Full</strong></td>
<td><strong>Full</strong></td>
<td><strong>Full</strong></td>
<td><strong>Full</strong></td>
<td><strong>Full</strong></td>
</tr>
<tr>
<td>7.2 Multi-level assignment/access control</td>
<td><strong>Full</strong></td>
<td><strong>Full</strong></td>
<td><strong>Full</strong></td>
<td><strong>Full</strong></td>
<td><strong>Full</strong></td>
</tr>
<tr>
<td>8. Interfaces to Other Tools</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.1 Inter-tool communications</td>
<td><strong>Full</strong></td>
<td><strong>Full</strong></td>
<td><strong>Full</strong></td>
<td><strong>Full</strong></td>
<td><strong>Full</strong></td>
</tr>
<tr>
<td>8.1.1 Interfaces to other tools?</td>
<td><strong>Full</strong></td>
<td><strong>Full</strong></td>
<td><strong>Full</strong></td>
<td><strong>Full</strong></td>
<td><strong>Full</strong></td>
</tr>
<tr>
<td>8.1.2 External Applications Program Interface available</td>
<td><strong>Full</strong></td>
<td><strong>Full</strong></td>
<td><strong>Full</strong></td>
<td><strong>Full</strong></td>
<td><strong>Full</strong></td>
</tr>
<tr>
<td>8.1.3 Support Open database system (standard query access)</td>
<td><strong>Full</strong></td>
<td><strong>Full</strong></td>
<td><strong>Full</strong></td>
<td><strong>Full</strong></td>
<td><strong>Full</strong></td>
</tr>
<tr>
<td>8.1.4 Import of existing data from various standard file formats</td>
<td><strong>Full</strong></td>
<td><strong>Full</strong></td>
<td><strong>Full</strong></td>
<td><strong>Full</strong></td>
<td><strong>Full</strong></td>
</tr>
<tr>
<td>8.1.5 Support Data Exchange Standard (SAP-233, XML,..)</td>
<td><strong>Partial</strong></td>
<td><strong>Partial</strong></td>
<td><strong>Full</strong></td>
<td><strong>Full</strong></td>
<td><strong>Full</strong></td>
</tr>
<tr>
<td>8.2 Intra-tool communications</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.2.1 Exchange of information between same-tool different installations</td>
<td><strong>Full</strong></td>
<td><strong>Full</strong></td>
<td><strong>Full</strong></td>
<td><strong>Full</strong></td>
<td><strong>Full</strong></td>
</tr>
<tr>
<td>8.2.2. Consistency/comparison checking between same-tool datasets</td>
<td><strong>Full</strong></td>
<td><strong>Partial</strong></td>
<td><strong>Full</strong></td>
<td><strong>Full</strong></td>
<td><strong>Full</strong></td>
</tr>
<tr>
<td>Tools:</td>
<td>IBM Rational RequisitePro (updated 10 Oct 04)</td>
<td>SLATE REquire 0.5 (UGS Teamcenter Product)</td>
<td>UGS Teamcenter 2005 (Requirements Management)</td>
<td>Cognition Cogit 5</td>
<td>Toelogic DOORS</td>
</tr>
<tr>
<td>-------</td>
<td>---------------------------------------------</td>
<td>---------------------------------------------</td>
<td>-----------------------------------------------</td>
<td>----------------</td>
<td>----------------</td>
</tr>
<tr>
<td>9. System Environment</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>9.1 Single user/multiple concurrent users</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>9.2 Multiple Platforms/Operating Systems?</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>9.3 Commercial vs. Proprietary database</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>9.4 Resource Requirements</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>9.4.1 Memory requirements</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>9.4.2 CPU Requirements</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>9.4.3 Disk space requirements</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>10. User Interfaces</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>10.1 Doing one thing while you are looking at another</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>10.2 Simultaneous update of open views</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>10.3 Interactive input/control of data</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>10.4 Which window standard do you follow?</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>10.5 Executable via scripts (recordable) or macros</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>10.6 Web browser interface?</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>10.7 Edit Undo Function Support</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
</tbody>
</table>
Conclusion

- What do you conclude from all this?

- Are the top tools really the same?

- If so, why not select the oldest tool with the largest customer base?
  - No one gets fired for doing that… right?
The trouble is, traditional systems are not doing their job.

**How often do we miss our targets?**

*Source: Aberdeen Group, June 2007*

- **Launch Targets**
  - Best (20% of Companies): 14%
  - Average (50% of Companies): 41%
  - Laggards (30% of Companies): 79%

- **Performance Targets**
  - Best (20% of Companies): 4%
  - Average (50% of Companies): 21%
  - Laggards (30% of Companies): 52%

- **Reliability Targets**
  - Best (20% of Companies): 15%
  - Average (50% of Companies): 26%
  - Laggards (30% of Companies): 47%
The Selection Criteria Needs to be Broader

- The INCOSE baseline criteria is a great starting point
- Numeric requirements, Voice of the Customer, Critical Parameter Management, and FMEA, Hazard and Requirement unification all need to be addressed.

**Traditional Requirements Management** has evolved from software-centric needs focused on careful documentation and traceability:
- Textual Reqs & Specs
- Textual V&V Tests
- Trace Matrices

**Modern Requirements Management** is driven by the additional goals of achieving lean execution of the full Product Development Process, across hardware, software and process:
- Market data analysis (segmentation, KJ, etc.)
- VOC/VOB/VOT/VOR assessment and prioritization
- Critical Parameter identification through HOQ and Hazard Analysis
- Design Concept Exploration and Selection (Pugh)
- System Tree/Functional Tree/Risk Tree Creation and Linkage
- Bottom-up FMEA, Mitigation and Control Plan Management
- 1st Eng. Principle Computation and Transfer Functions
- Critical Parameter Flowdown and Design Prediction Flowup
- Statistical Variation Target Assessment/DCM Reporting
- Design of Experiments and Test Management
- Product Cost Estimation and Initiative Management
Would You Benefit from Cockpit?

- We recommend that a full evaluation includes a lot more than baseline functions.
- Look at your development deliverables and add additional items.

<table>
<thead>
<tr>
<th>Requirement/Specification</th>
<th>IMP.</th>
<th>DOORS</th>
<th>REQPRO</th>
<th>COCKPIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Textual Requirement/Specification Definition</td>
<td>100</td>
<td>10</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>Textual Validation Tests</td>
<td>70</td>
<td>10</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>Broad PDP Trace Matrices</td>
<td>100</td>
<td>6</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>Market data analysis (Segmentation, KJ, etc.)</td>
<td>80</td>
<td>3</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>VOC/VOB/VOT/VOR assessment and prioritization</td>
<td>80</td>
<td>2</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>Critical Parameter identification through HOQ &amp; Hazard Analyses</td>
<td>90</td>
<td>0*</td>
<td>0*</td>
<td>8</td>
</tr>
<tr>
<td>Design Concept Exploration and Selection (Pugh)</td>
<td>50</td>
<td>0*</td>
<td>0*</td>
<td>8</td>
</tr>
<tr>
<td>System Tree / Functional Tree / Risk Tree Creation and Linkage</td>
<td>100</td>
<td>0*</td>
<td>0*</td>
<td>9</td>
</tr>
<tr>
<td>Bottoms-up FMEA, Mitigation, and Control Plan Management</td>
<td>80</td>
<td>0*</td>
<td>0*</td>
<td>7</td>
</tr>
<tr>
<td>1st Eng. Principle Computation and Transfer Functions</td>
<td>60</td>
<td>0*</td>
<td>0*</td>
<td>10</td>
</tr>
<tr>
<td>Critical Parameter Flow-down and Design Prediction Flow-up</td>
<td>90</td>
<td>0*</td>
<td>0*</td>
<td>10</td>
</tr>
<tr>
<td>Statistical Variation Target Assessment / Scorecard Reporting</td>
<td>60</td>
<td>0*</td>
<td>0*</td>
<td>9</td>
</tr>
<tr>
<td>Design of Experiments and Test Management</td>
<td>90</td>
<td>5</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>Product Cost Estimation and Initiative Management</td>
<td>80</td>
<td>0*</td>
<td>0*</td>
<td>8</td>
</tr>
</tbody>
</table>

* Function not supported
Would You Benefit from Cockpit?

- If you have a Requirements Management system, you may observe the following:
  - Fewer people than expected are willing to use it
  - Updating data has been assigned to a small group of people who can handle the complexity of the tool
  - You spend enormous amounts of time administering links and traces

- Regarding ease of use and deployment:

<table>
<thead>
<tr>
<th>Feature</th>
<th>IMP</th>
<th>DOORS</th>
<th>REQPRO</th>
<th>COCKPIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intuitive user interface for easy adoption by all project personnel</td>
<td>100</td>
<td>4</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>100% Browser based for ALL actions (read/write)</td>
<td>90</td>
<td>2</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>Ease of Customization to Meet SOP Template Deliverables</td>
<td>100</td>
<td>2</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>Ease of integration with SolidWorks, Goldfire, Windchill</td>
<td>70</td>
<td>2</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>920</strong></td>
<td><strong>850</strong></td>
<td><strong>3430</strong></td>
<td></td>
</tr>
</tbody>
</table>
Cockpit – A Unified Model for Product Development

- What, specifically, is different about Cockpit?
Cockpit – A Unified Model for Product Development

- Cockpit provides a superset of functionality—handling the needs of hardware, software, and process groups.
Cockpit – A Unified Model for Product Development

- Cockpit runs entirely in your Web browser and users actually enjoy using it.
- Everything is connected and everything has a Dashboard.
Cockpit – A Unified Model for Product Development

- Cockpit is a Unified Model for capturing, tracing, and managing all the product development data of your projects.

**Unified Model**

Where the “lines” are even more important than the “circles.”
Cockpit – A Unified Model for Product Development

- Just like email, everyone is plugged in. Changes in the Requirements Document are propagated to the other departments and change notifications are directly visible.
Cockpit – A Unified Model for Product Development

VOC, RM, Risk and Critical Parameter Functions are all needed

It’s all connected in Cockpit!

- Flowing-down customer requirements and Risk Mitigation into product & process design (specs & drawings)
- Flowing-up process capability & predicted performance up to customer requirements

<table>
<thead>
<tr>
<th>Proposal</th>
<th>Definition</th>
<th>Development</th>
<th>Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Market Spec</td>
<td>Product Spec</td>
<td>Design Freeze</td>
<td>Design Verification</td>
</tr>
</tbody>
</table>
Cockpit – A Unified Model for Product Development

For example, VOC management is critical to delivering a market success.

Driving Product Development with Critical Parameters

Cognition Delivers Active Requirements Management for the Full Product Lifecycle

PLM Integration/Product Definition

INTEGRATING THE VOICE OF THE CUSTOMER ACROSS THE FULL DEVELOPMENT CYCLE

In general, requirements management readily deals with a large amount of text that precisely defines technical requirements and detailed product specifications. But Cognition extends the capability to dynamically manage the handling of the original thought or need that is expressed by the customer or the members of a marketing organization.

At this early stage, several industries must track the source of a requirement, its justification, and any discussion that led to its formulation. Later, if a change has to be made and the requirement needs to be re-evaluated, it will be necessary to remember why the requirement was expressed as it was and the context of the discussion and marketing approaches considered.
Cockpit – A Unified Model for Product Development

- Capturing and analyzing your market research data is fully supported in Cockpit. You can import interview survey data and look for market segment trends.
Cockpit – A Unified Model for Product Development

- Cockpit’s affinity grouping tool is an electronic rendition of sticky notes. Cockpit creates them, and your team groups them by similarity. In the end, marketing has a much better understanding of the dominant customer needs—and they also have a great tool for communicating with engineering.
Prioritizing customer needs is also an important part of marketing’s communication with engineering. Cockpit templates offer a number of voting tools to do this.
Cockpit – A Unified Model for Product Development

- Flowing VOC requirements down methodically into lower level System Requirements and further into Sub System Requirements is fully supported with “best practice” methods. Shown here is a typical QFD approach.
Cockpit – A Unified Model for Product Development

- Pugh concept exploration using Cockpit’s templates is very convenient. Here, four concepts are being evaluated against System Requirements criteria defined earlier. And the weighting of the criteria came from the QFD template using VOC importance.
Next, if you want to configure a “master trace” of your own design, you can click on the details page of any document and define the columns the way you want them.
Cockpit – A Unified Model for Product Development

- And this is how it looks. You can then right click and create a template out of your settings and other engineers can use it at any time in their project.
Cockpit – A Unified Model for Product Development

In many projects, more than 50% of the requirements are mitigating risks of some kind. Managing risks and requirements separately in different tools is not only difficult, but the source of many compliance problems. In Cockpit, they are completely interconnected.
Cockpit – A Unified Model for Product Development

- Cockpit supports unlimited levels of risk decomposition. Hazards, harms, hazardous situations, etc. are completely supported, and libraries of these items can be stored and reused from project to project.

---

<table>
<thead>
<tr>
<th>Level 1</th>
<th>Level 2</th>
<th>Level 3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Functional Requirement</strong></td>
<td><strong>Potential Failure Mode(s)</strong></td>
<td><strong>Potential Effect(s)</strong></td>
</tr>
<tr>
<td>RQMT 1</td>
<td>FAILURE 1</td>
<td>ASSY FAILURE 1</td>
</tr>
<tr>
<td>RQMT 2</td>
<td>ASSY FAILURE 1</td>
<td>FAILURE 1</td>
</tr>
<tr>
<td>RQMT 3</td>
<td>SUB-ASSY FAILURE 1</td>
<td>ASSY FAILURE 1</td>
</tr>
</tbody>
</table>
Cockpit – A Unified Model for Product Development

- And all risks are connected to their mitigations and resulting controls (requirements). Residual risk is calculated automatically and FMEA, Hazard Analysis, FTA reports for Design, Process, Use, etc. are fully supported.
Cockpit – A Unified Model for Product Development

- In addition to risk mitigation, a large number of requirements exist because they are critical to quality or product performance.

- For a Requirements Management system to be helpful to the entire organization, it must support numeric requirements as well as allow the analysis of their statistical variation and predicted compliance. Cockpit is the only tool supporting this, yet it is critical to achieving predictable quality.

**VOC, RM, Risk and Critical Parameter Functions are all needed**

- Flowing-up process capability & predicted performance up to customer requirements
There Are Two Approaches to Design

**Normal Approach:**
A simple margin calculation does not tell us enough about the design.

**Statistical Approach:**
A simple, yet statistical approach allows us to predict short and long term product performance and manufacturing capability.

“Variation is the root of all evil.”
– DFSS Principle
Corporate-Wide RM System Must Provide More

- Critical Parameter Management (CPM) is the analytic capability of predicting product performance by statistically assessing the impact of lower-level design variation all the way up to customer experience.

CPM exposes design risks and opportunities

And savings opportunities (over-design)

CPM Exposes Performance Risks

© Cognition Corporation 2014
Cockpit – A Unified Model for Product Development

For example, the requirement “shall be between .09 and .17 inches”. Yet, the current predicted value is .1325 +/- .05026—which is a problem. If we know this early, we can fix it more cost effectively.
So, in addition to the “customer target” for each requirement, there is also an assessment of the current specified or predicted “design value.”

Cockpit computes this value throughout the development process using transfer functions, which are then reported in scorecards or stoplight charts used to make decisions and tradeoffs.
Cockpit – A Unified Model for Product Development

- Cockpit then uses its parent/child linkages to predict which lower level sub-requirements most contribute to the problem, as well as which higher level requirements will be affected by a resulting design change.
Cockpit – A Unified Model for Product Development

- Time consuming and expensive “design of experiments (DOEs)” are often performed in order to predict performance.

- Cockpit supports electronic DOEs and Monte Carlo analyses directly.

- It also links with Excel, Matlab, Visio and other tools. Ask your team if they do these analyses.
Cockpit – A Unified Model for Product Development

- Cockpit allows engineers to see when their designs are getting more robust. Similarly, they show when items are over-designed and therefore more costly than necessary.
Cockpit – A Unified Model for Product Development

- So, whether you’re asking Risk, VOC, CPM, V&V, or Requirement related questions, Cockpit connects them all.

- Its Dashboard feature allows engineers and managers to see everything that is connected to any particular item, including verification tests, risk mitigations, parent and sub requirements, action items, notebook entries, and where-used documentation.

- With this info, impact analysis is made a lot quicker…
Cockpit – A Unified Model for Product Development

- Finding data is important also.
- Cockpit incorporates both hardware and software support for fast searching.
- Users can search for items in Cockpit in the same way they search for lawn mowers.
Cockpit – A Unified Model for Product Development

For example, let’s say you want to find all requirements for a particular subject:

Google search for Hotel Automated Parking System:

SR0498: Not more than 100 meter walk - Statistical Analysis Of Critical Parameters
reqs name: Not more than 100 meter walk id: SR0498 type: SR num: 498 owner: David Cronin
author: David Cronin state: In Process - Editable status: modified: 2009/08/13 ...

Hotel Automated Parking System (INCOSE) - Crawled: - 16k - Preview Where-used Similar Graphical

NOTE: Google Search Appliance is optional and NOT provided by Cognition.
Cockpit – A Unified Model for Product Development

- “Everything you put in Cockpit will be automatically crawled and indexed each night. (with a supplied Google Appliance on your network, not public)”

- “Every piece of research, every interview, every VOC, every requirement, every specification, every test, every protocol and V&V plan, every risk, hazard, mitigation, PowerPoint, Visio,

- Excel, meeting minutes, marketing document, business plan, characterization study, etc.”

- “The next morning and for months and years thereafter, you will have sub-second retrieval of all of it.”

*NOTE: Google Search Appliance is optional and NOT provided by Cognition.*
Lastly, managing Meetings Minutes and Action Items is also unified inside Cockpit. We know you have other tools for this, but if you use Cockpit, you can have direct linkages between your data and the meetings/actions that discuss them.
In the end…

- Functions cannot be isolated; department tools cannot be isolated; and your requirements cannot be isolated.

- Cockpit is the one system that brings them all together – satisfying the needs of your hardware, software and process development groups.

<table>
<thead>
<tr>
<th>Need</th>
<th>Cockpit</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOC Management</td>
<td>X</td>
</tr>
<tr>
<td>Requirements Management</td>
<td>X</td>
</tr>
<tr>
<td>Test Management</td>
<td>X</td>
</tr>
<tr>
<td>Critical Parameter Mgt</td>
<td>X</td>
</tr>
<tr>
<td>Risk Management</td>
<td>X</td>
</tr>
<tr>
<td>Cost Management</td>
<td>X</td>
</tr>
</tbody>
</table>

Where, ease of use and ease of deployment are key:
- One environment
- One database
- One interface
- One price
- One installation
- One partner
Our Conclusion

- Companies need a Single Unified Model that handles all of it – a system for capturing, tracing, and managing all the product development data of their world.

Where the “lines” are even more important than the “circles.”
Cockpit - A Unified Model For Product Development

- Users
- Finance
- Manufacturing
- Supply Chain
- Cost Estimates
- Quality
- Standards/Regulations
- Design
- Subject Matter Expert
- Price Control
- Risk
- Systems Engineering
- Marketing
- Development Project
Cockpit – A Unified Model for Product Development

- With the Unified Model:
  - Everything is connected
  - Everything is templated
  - Everything is indexed

- We address:
  - Right Product
    - Compliance
    - Time to Market
  - Quality & Performance
Cockpit – A Unified Model for Product Development

Start with an evaluation!

Your users will embrace it like nothing you’ve seen before.

Cognition Corporation
213 Burlington Road, Suite 109
Bedford, MA 01730
http://www.cognition.us

sales@cognition.us
781-271-9300 x260