A new kind of music:
making the move from DOORS Classic to Jazz / DOORS NG

Jeremy Dick – **integrate**
Parham Vasaiely – Airbus
Jason Wilson – World-Wide Clinical Trials

Innovate2013
The IBM Technical Summit

objectives

- to share some customer experiences relating to making the transition from DOORS Classic to DOORS NG
  - challenges and benefits

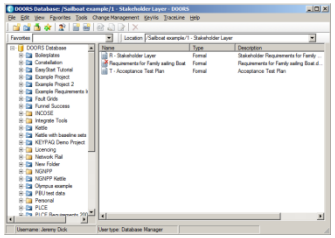
and by doing so,

- to give other organisations the confidence to do the same

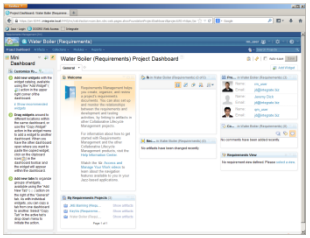
key message

moving from DOORS Classic to DOORS NG involves a shift in thinking

- user experience:



DOORS Classic client



Jazz in browser

- database concepts:

project
folder
module
object

lifecycle project
project area
folder / module / collection
artefact

- integration:

point-to-point solutions
replicated data
synchronisation

seamless data and process integration
traceability across tools
Jazz platform
3rd-party tool integration by open standards

key message

it takes some time and effort to make the transition in your thinking, but once made, the benefits of DOORS NG over DOORS Classic are quickly apparent

- more refined information models:
  - same attributes for all objects
  - attribute definitions local to module
  - views local to module

  - user-definable artefact types
  - attributes per artefact type
  - attributes and views global to project

- dynamic traceability:
  - static links
  - static traceability columns
  - DOORS links v. external links

  - mouse-over pop-up panels
  - dynamic traceability columns
  - seamless linking to other tools

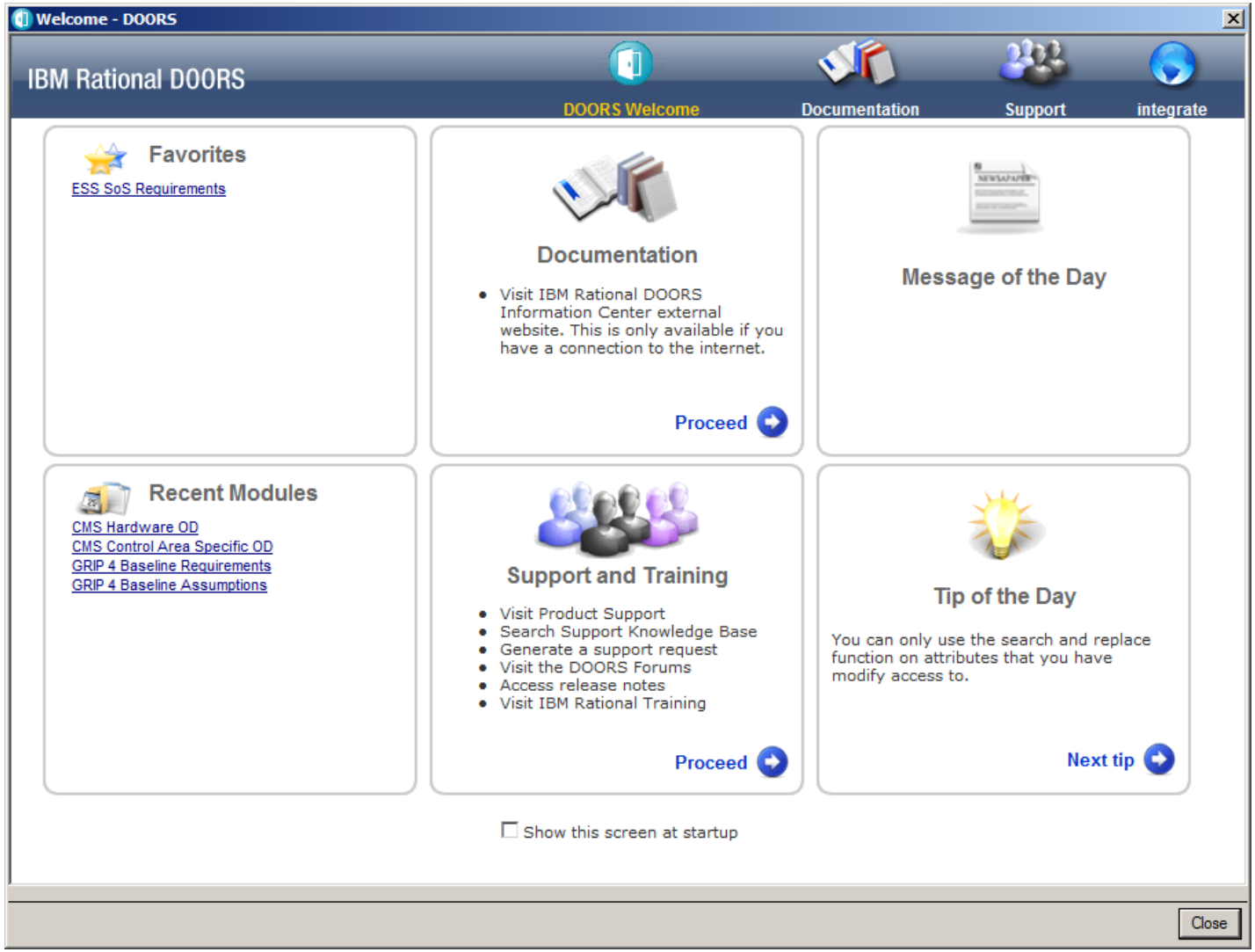
- document generation:
  - partial document generation
  - manual document compilation

  - lifecycle document using RPE

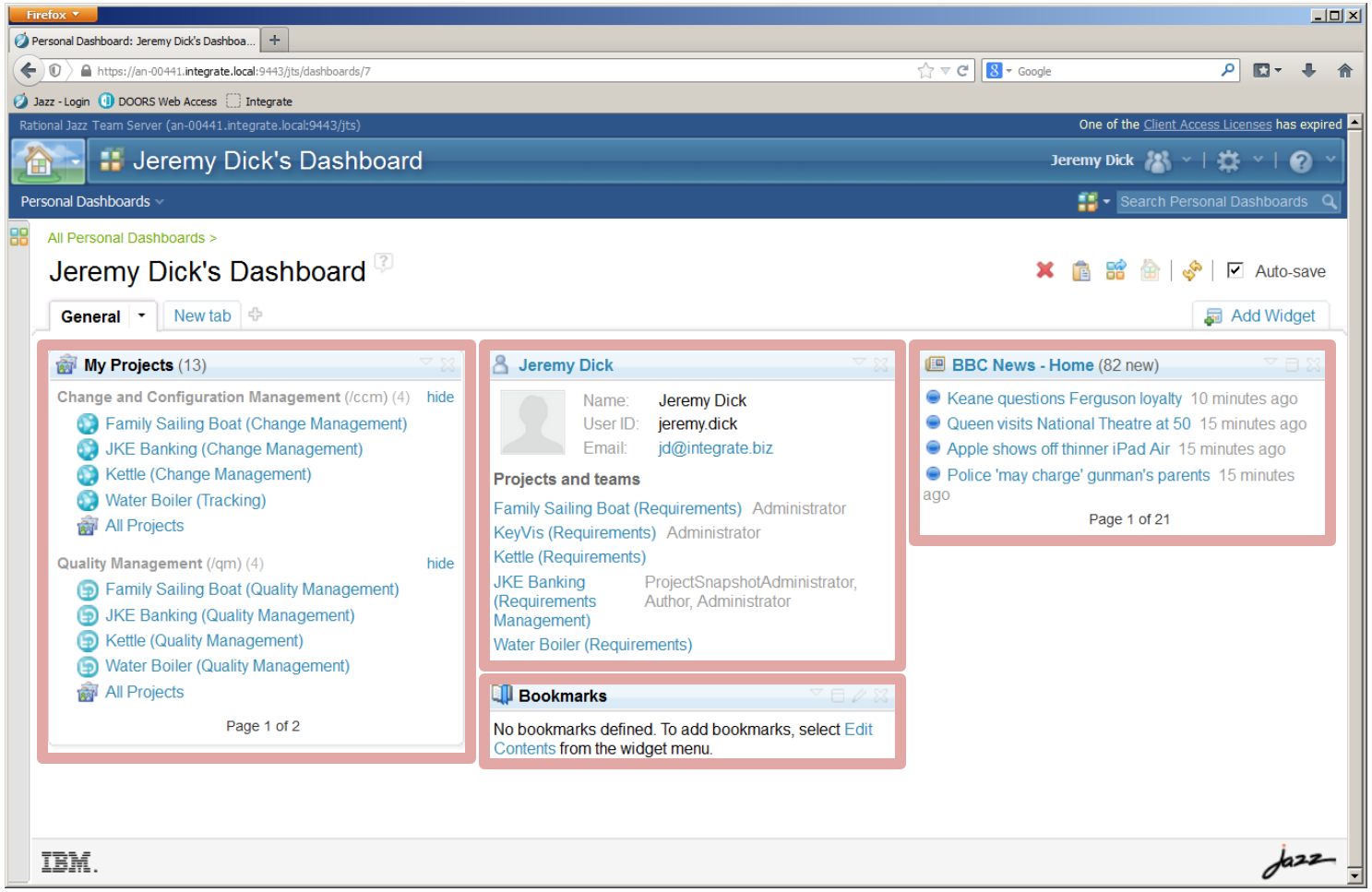
Jazz concepts I had to understand

- dashboards that carry widgets
  - every user has a dashboard
  - every project area has a dashboard
  - there are mini-dashboards on most screens
  - wide range of widget purposes: communication, status, metrics, applications
  - opportunity for customisation
- life-cycle projects v. project areas
  - life-cycle projects are made up of project areas
  - the number of project areas depends on the scope of the project
- artifacts
  - artifacts = objects

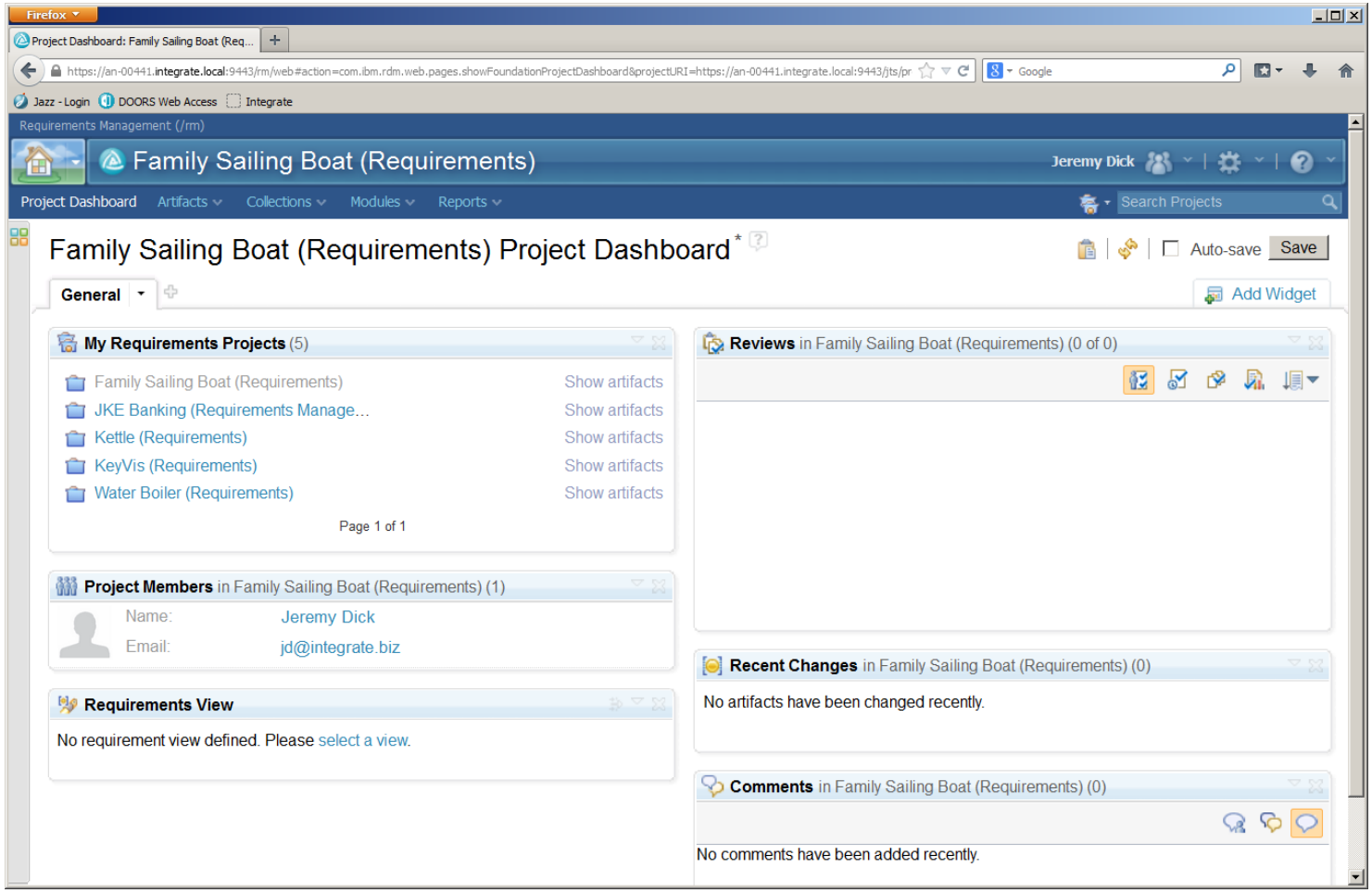
DOORS Classic welcome screen



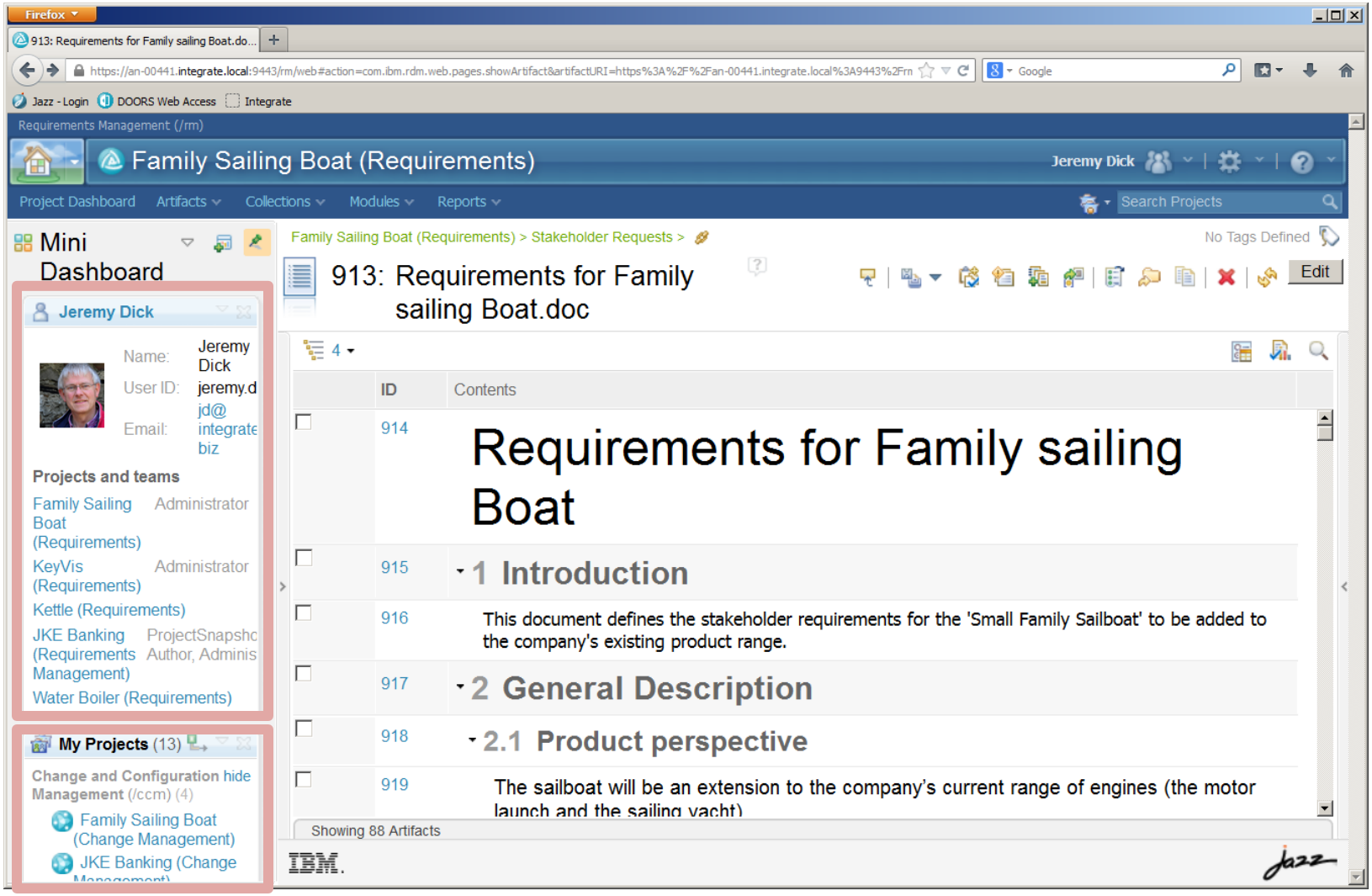
understanding Jazz dashboards and widgets



dashboards for everything
- users / life-cycle projects / project areas / …



mini dashboard – artifact-level widgets



/ˈwɪdʒɪt/ *noun informal*

- a small gadget or mechanical device.
- (in some beer cans) a plastic device which introduces nitrogen into the beer, giving it a creamy head.
- COMPUTING: an application, or a component of an interface, that enables a user to perform a function or access a service.

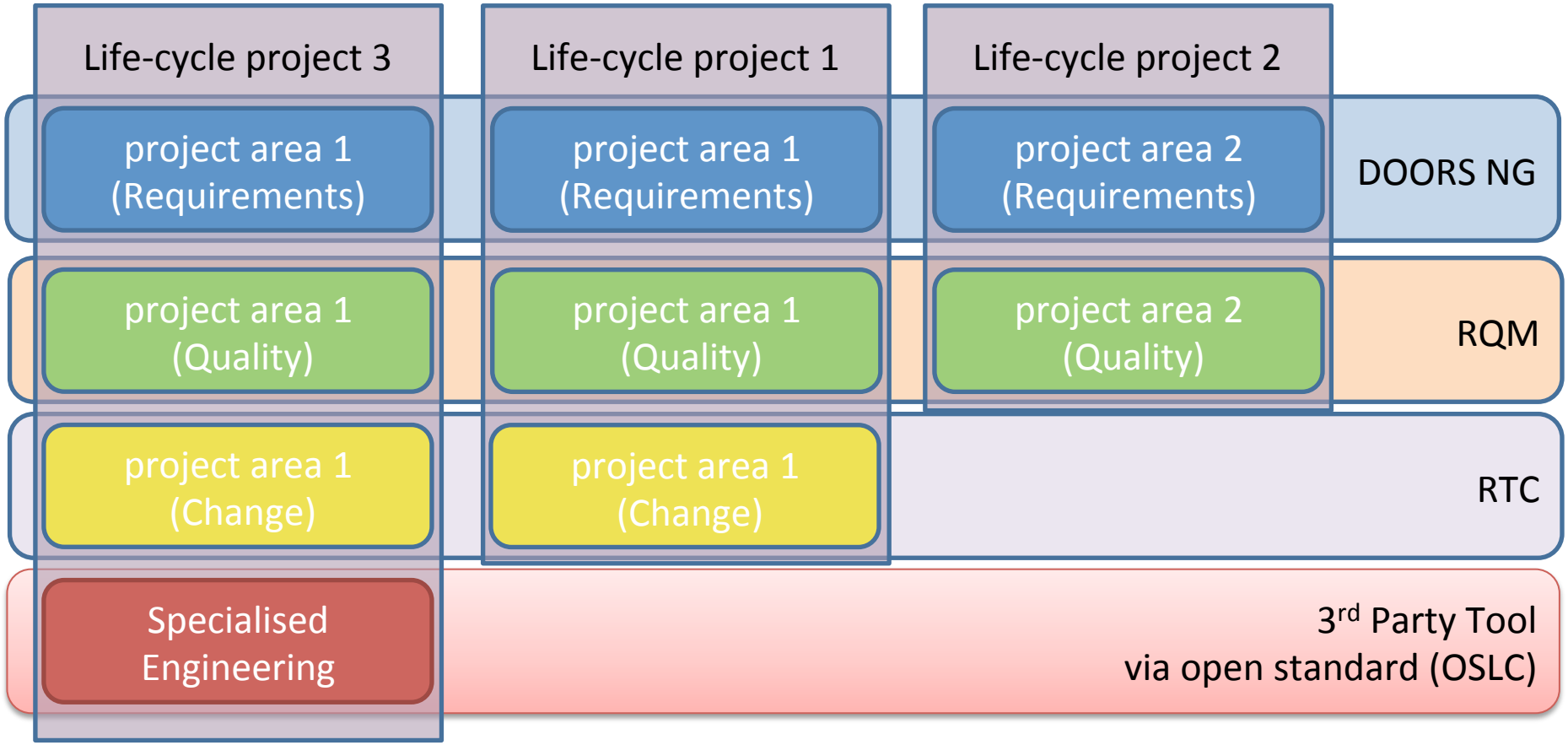
- about 100 widgets available as standard
  - navigation aids
  - reports and metrics
  - external sources
- offers extensibility and customisation
  - develop new widgets (requires Java/Java script expertise)

/ˈɑːtɪfakt/ *noun*

- An object produced or shaped by human craft, especially a tool, weapon, or ornament of archaeological or historical interest.
- COMPUTING: tangible by-product produced during the development of software.

- the things you work with:
  - requirements, user stories, test cases, defects, tasks, builds, configurations, comments, ….
- each application defines the types of artifact relevant to their sphere of operation
- in the case of DOORS NG, you can create new artifact types
- artifacts have:
  - a type (determining content)
  - a format (determining presentation)
  - an identifier unique across the whole database

understanding projects



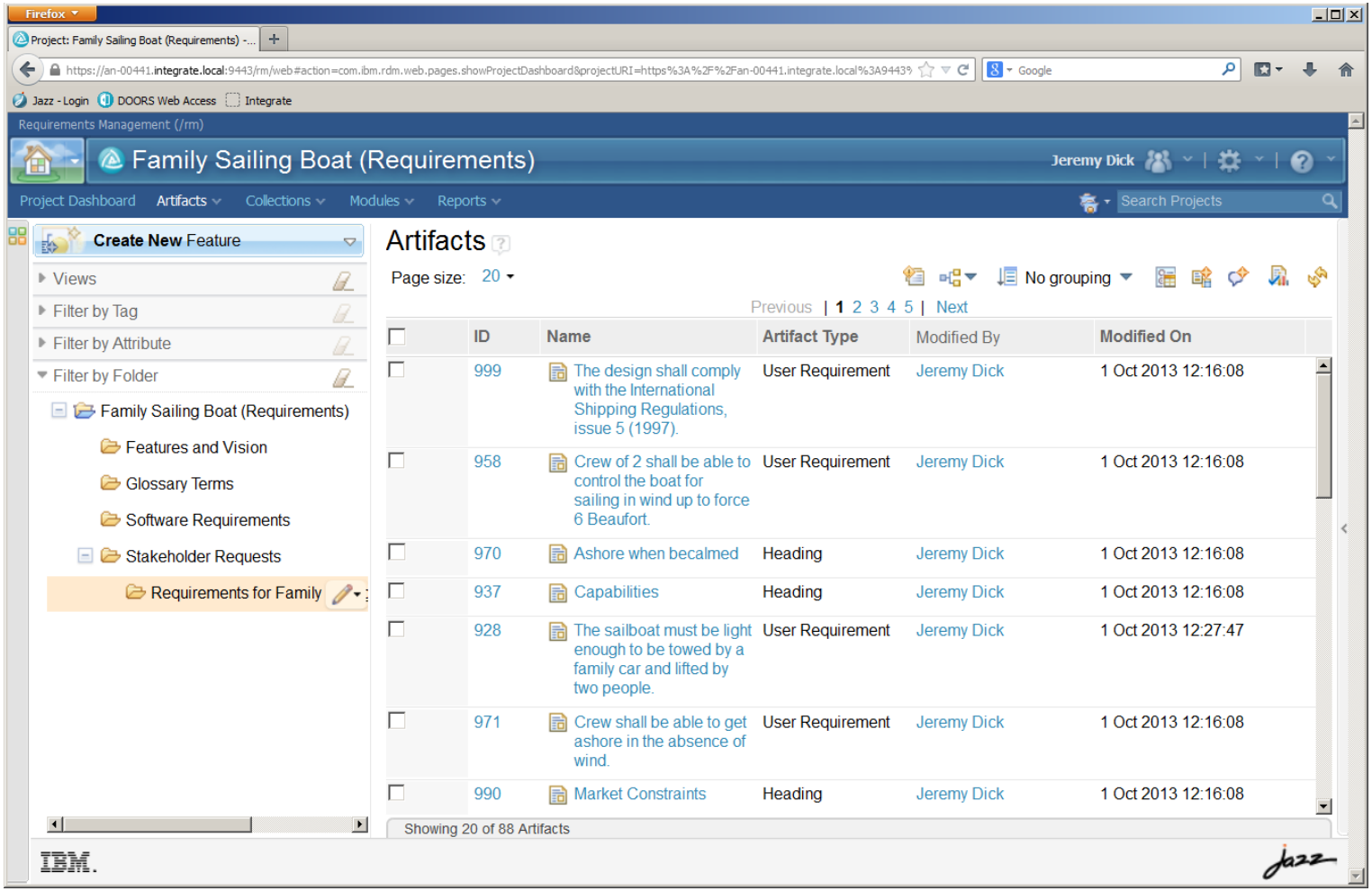
DOORS NG concepts I had to understand

- folders, collections and modules
  - artifacts live physically in folders
  - artifacts may be placed into any number of collections
  - a module is a collection presented in a particular format

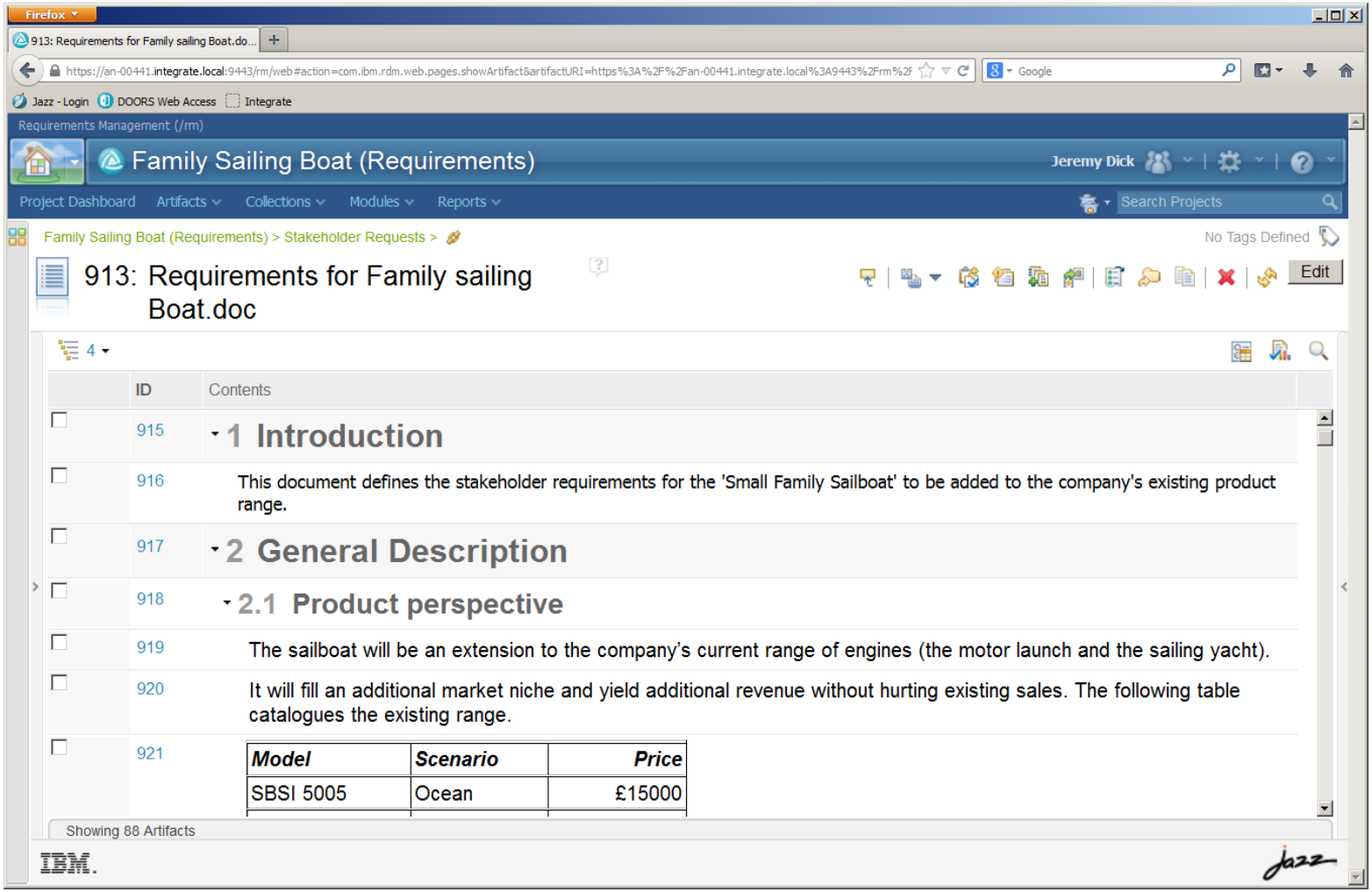
understanding folders, collections and modules

- folders are where artifacts live
  - organised into a hierarchy of folders
  - each artifact lives in exactly one folder
- collections are artifacts that may contain other artifacts
  - *e.g.* the set of requirements to be tested on a particular test rig
  - each artifact may live in any number of collections
  - collections cut across the folder structure
  - (a foundation for versions, variants and reuse)
- modules are collections formatted to look like classic DOORS
  - “module” is an artifact format
  - headings and hierarchical structure

folders of artifacts



modules – familiar territory

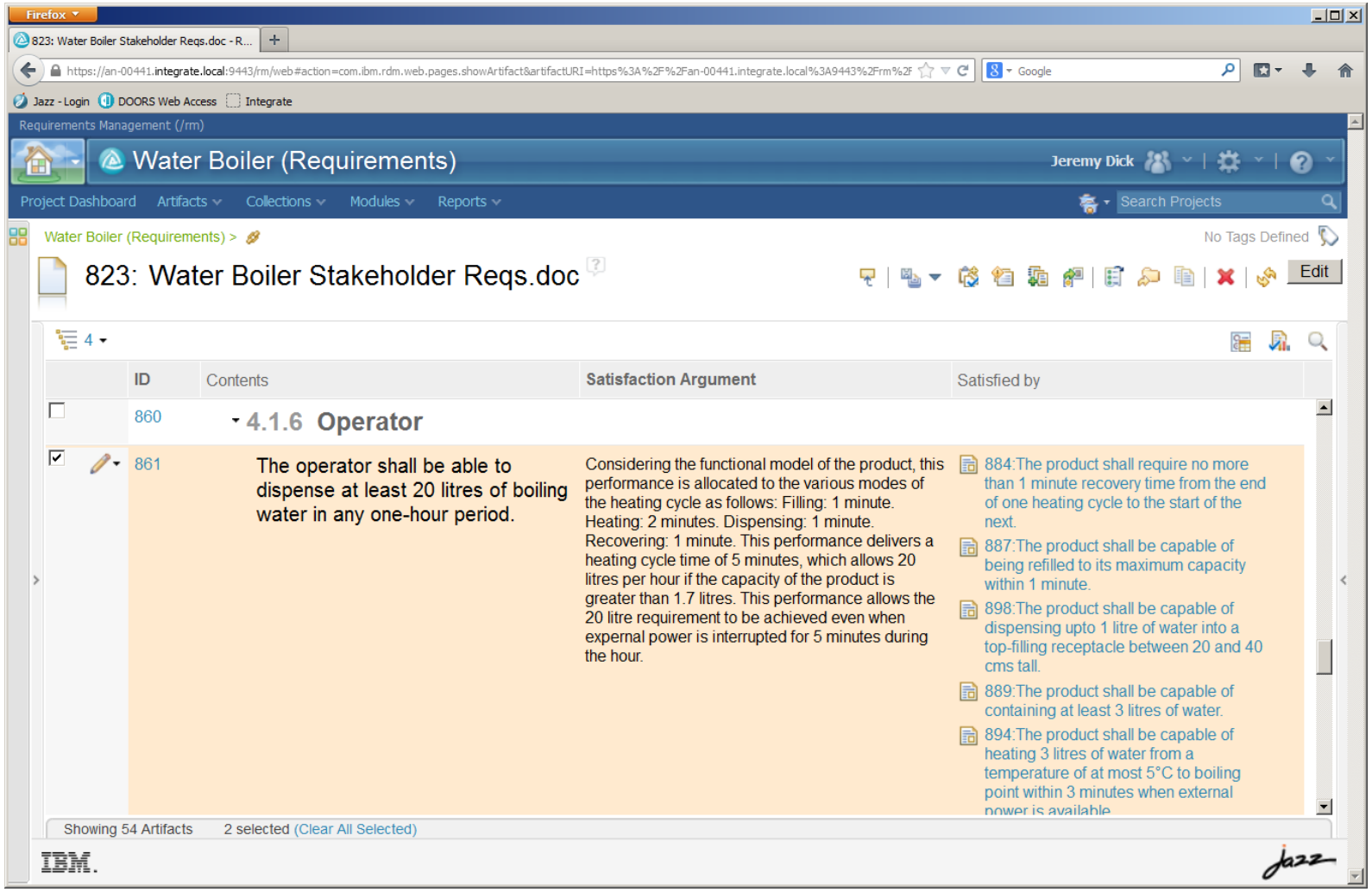


filtering and views – familiar territory

- folders, collections and modules can all have user-defined
  - filters
  - views (that can be named and saved)

- a view is an arrangement of columns
- columns can show attributes and traceability

traceability reports



what is better in DOORS NG

- attributes / views management
  - attributes are applicable to artefact types
  - attributes are global to the project
  - views can also be global

- traceability columns are dynamic
  - linked artefacts displayed as navigable links
  - “rich hover” gives linked artefact details
  - links can be created and deleted from the traceability column

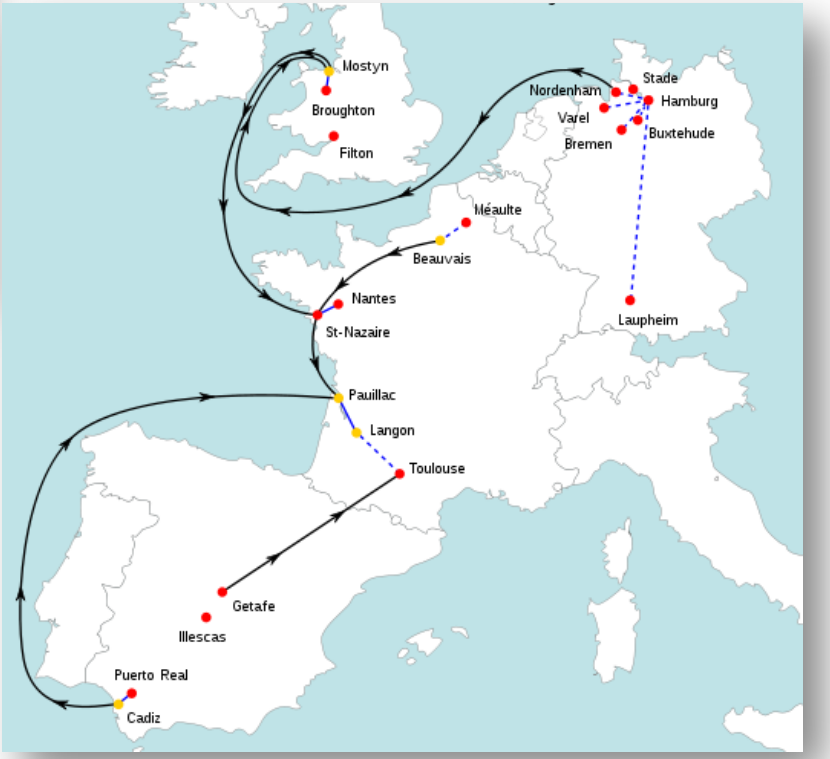
dynamic traceability columns

Airbus

CHALLENGES OF COMPLEX SYSTEMS ENGINEERING

Airbus key challenges for engineering lifecycle management – A380 example

- complex products
- safety-critical systems (Certification e.g. DO-178B)

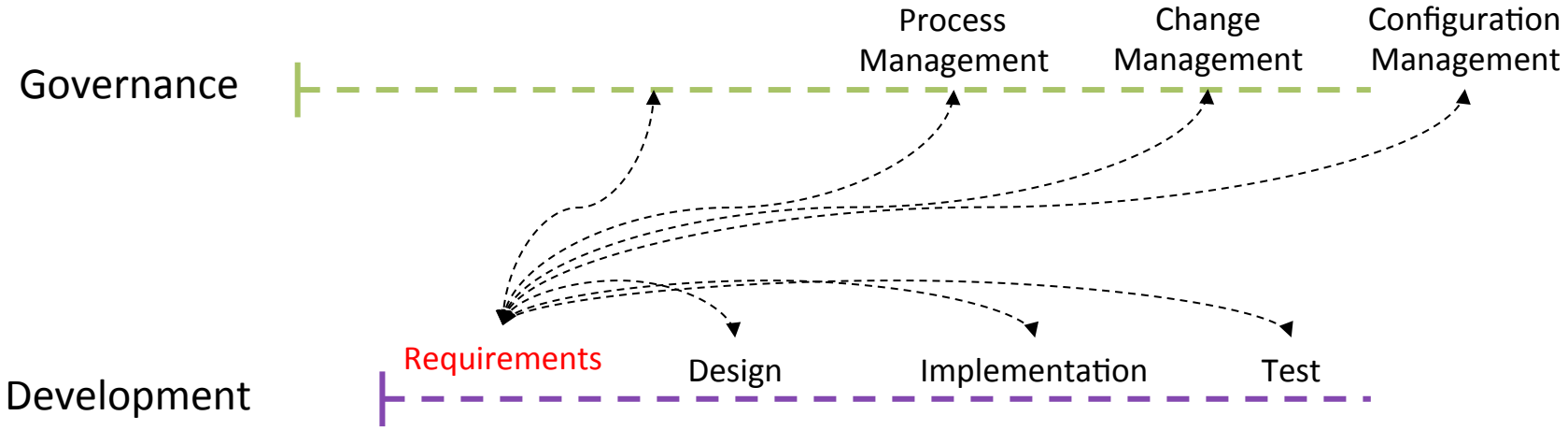


- geographically distributed engineering teams
- complex IT infrastructure
- extended enterprise

safety-critical systems
→ managing DO-178B and DO-254 compliance

- one of the important activities in both standards is the requirements capturing and tracking throughout the design and verification process.

- using an integrated engineering environment for systems and software engineering



a new kind of openness

DOORS Classic is very “open” through ability to customise

DOORS NG is very “open” through standard integration

- the need to customise reduced through open integration and standard APIs
- ability to customise through non-proprietary languages
  - provision of custom widgets within Jazz
  - provision of OSLC-capable applications

Open Services for Lifecycle Collaboration (OSLC)

- OSLC is an open and scalable approach to lifecycle integration
- simplifies key integration scenarios across heterogeneous tools

The OSLC Core specifies the primary integration techniques for integrating lifecycle tools, such as communication protocols and basic data representation
The OSLC domain *workgroups* specify additional vocabulary specific to their lifecycle domain, but do not add new protocols

Architecture of the Web

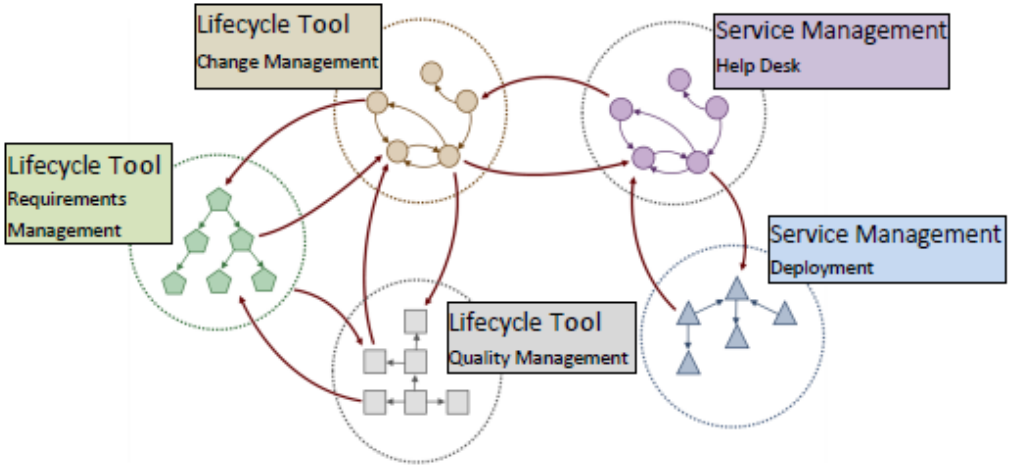
Linked Data

Standard Interfaces

Increased traceability

Increased reuse

Decreased maintenance costs



migration options

- don’t integrate or migrate
  - stick with what you’ve got
  - (maybe because high investment in custom DXL)
  - OK, because DOORS Classic will live on!

- integrate DOORS Classic with Jazz tools
  - stick with what you’ve got
  - use OSLC integration with RQM and RTC
  - benefit from life-cycle tools

- migrate from DOORS Classic to DOORS NG
  - one-off migration (using ReqIF)
  - benefit from the new features

summary of key message

a new kind of harmony

- although moving to DOORS NG involves thinking in new ways, the benefits of DOORS NG over DOORS Classic are quickly apparent

QUESTIONS?

Innovate2013
The IBM Technical Summit

summary of key message

a new kind of harmony

- although moving to DOORS NG involves thinking in new ways, the benefits of DOORS NG over DOORS Classic are quickly apparent

A BIG THANKS TO …

Parham Vasaiely – Airbus

Jason Wilson – WWCT

Innovate2013
The IBM Technical Summit