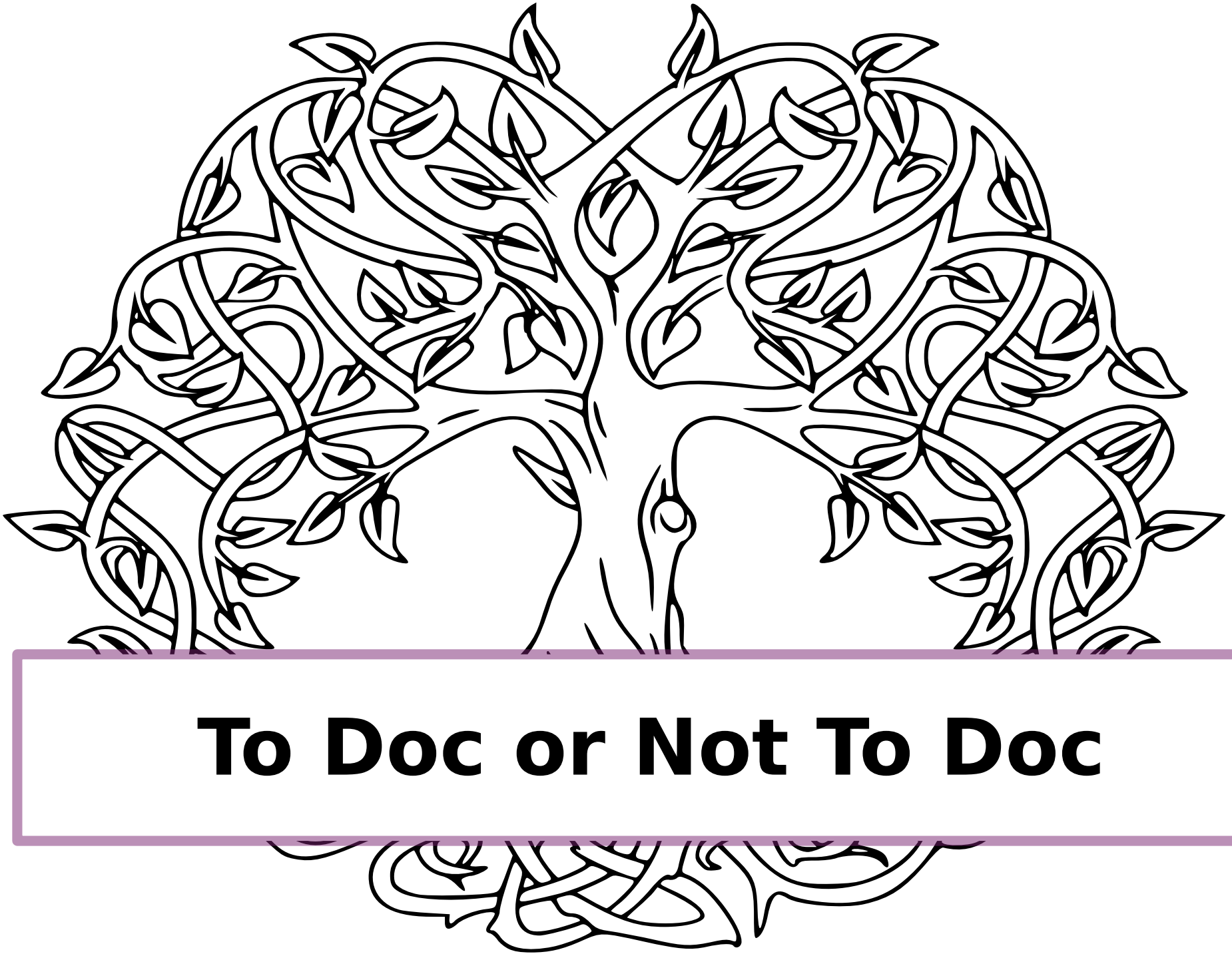


To Doc or Not To Doc

Chi sono

- Nicola.Pedot@informatelier.it

Software Doc

- Quale problema vogliamo risolvere con la documentazione?
- Quale problema stiamo introducendo con la documentazione?

- parliamone ....

InformAtelier

Un giorno

Senza intermediari!
Senza traduzioni!
Semplicemente diremo al computer quello che vogliamo… esattamente.

Siii , ma

oggi non è proprio così!

SCRUM

Scrum is a management framework that is becoming increasingly more common in the software industry. Where traditional methods focus on staying on track, Scrum is aimed at – like other **Agile methods** - delivering business value. Scrum provides a platform for people to work together effectively and relentlessly makes visible every problem that gets in its way.

Manifesto for Agile Software Development:
Individuals and interactions *over* processes and tools
Working software *over* comprehensive documentation
Customer collaboration *over* contract negotiation
Responding to change *over* following a plan

The essence of Scrum is:
- The team is given clear goals
- The team organises itself around the work
- The team regularly delivers the most valuable features
- The team receives feedback from people outside it
- The team reflects on its way of working in order to improve
- The entire organisation has visibility into the team's progress
- The team and management honestly communicate about progress and risks

Product Owner (PO)

Roles

Responsible for Product backlog creation and prioritizing

- Creates Product backlog
- Prioritizes Product backlog
- Manages Releases
- Describes features to the Team
- Accepts or rejects work results

Scrum Master (SM)

Roles

Responsible for team to follow scrum values

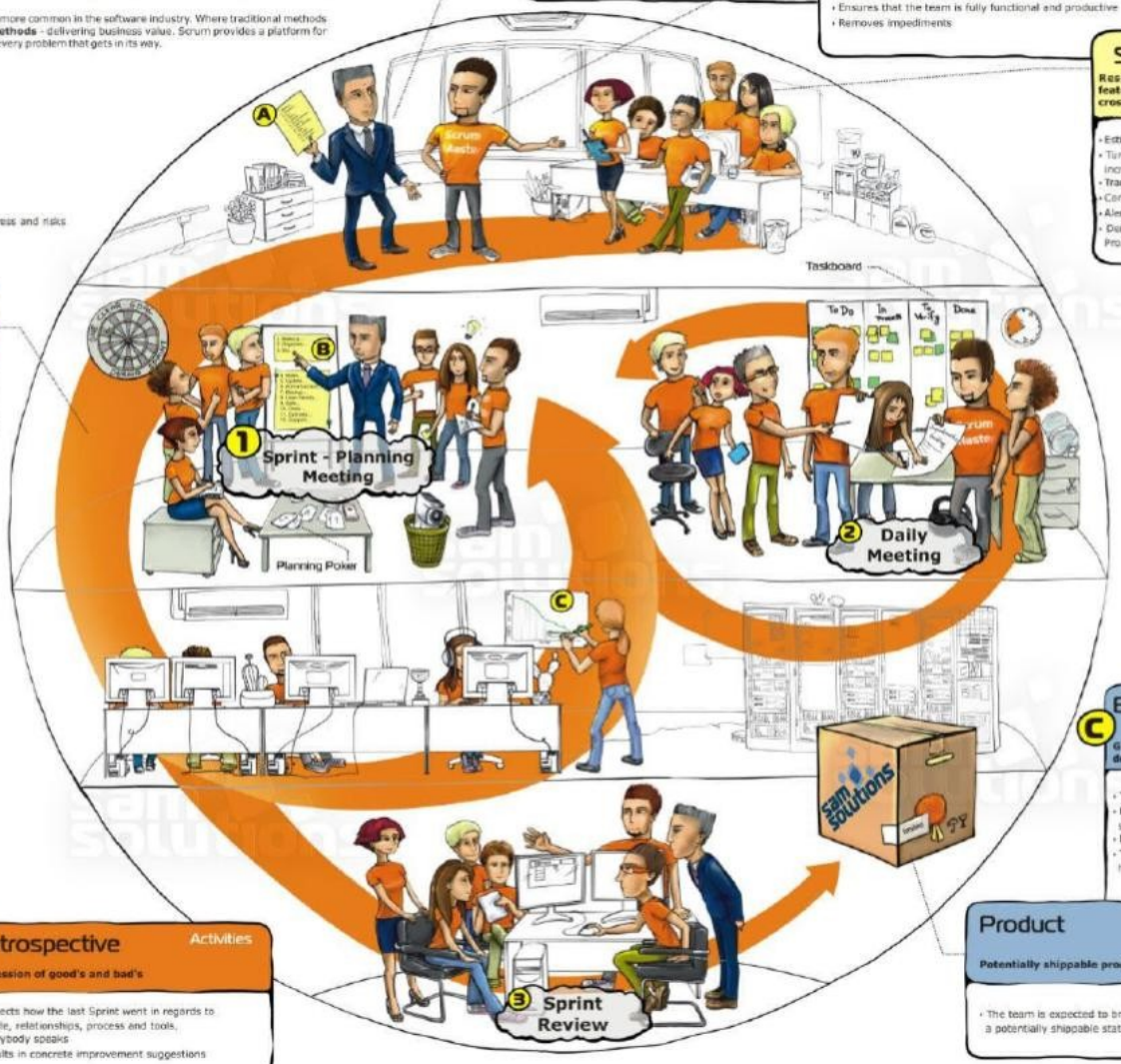
- Runs Sprint Planning and Daily Scrum meetings
- Ensures that the team is fully functional and productive
- Removes impediments

Scrum Team

Roles

Responsible for estimation and implementing features. Highly motivated, self organized and have cross functional skills

- Estimates size of Sprint backlog items (Planning Poker)
- Turns Sprint backlog list into Potentially shippable product increment
- Tracks work progress every day (Daily Scrum Meeting)
- Communicates with product owner regularly
- Alerts when there are problems
- Demonstrates Potentially shippable product Increment to Product owner



Sprint

Activities

2–4 week period of work on new features of Product

- Team uses appropriate best engineering practices during sprint
- Team delivers something after each sprint

1 Sprint Planning Meeting

Activities

Selecting, discussion, and estimation of features for current sprint

- Selecting, analyzing, and estimating Product Backlog for Sprint
- Product owner describes the details of the features to the Team
- Product owner answers the questions from the Team
- The Team plays Planning Poker and estimates the features
- All team members commit to them

2 Daily Scrum Meeting

Activities

Status of work progress meeting

- Same time, same place every day, lasts 15 minutes.
- Everybody answers the 3 questions:
  1. What have I done?
  2. What am I going to do?
  3. What problems do I have?
- Team members address each other not Scrum Master

Sprint Review

Activities

Demonstration of implemented features on working product

3

- Demo is done after each sprint
- All stakeholders and other teams can be invited to the demo
- Demo shows working product
- Product owner makes a decision whether team has achieved the goal of the sprint

Retrospective

Activities

Discussion of good's and bad's

- Inspects how the last Sprint went in regards to people, relationships, process and tools.
- Everybody speaks
- Results in concrete improvement suggestions

A Product Backlog

Artifacts

A list of features to be implemented in Product

- Contains User Stories, Issues, Bugs or Technical Tasks
- All the items are prioritized by Product Owner
- All the items are regularly updated by Product Owner

B Sprint Backlog

Artifacts

A list of features to be implemented in current Sprint

- Features are selected from Product Backlog by Team on the basis of their priorities
- The features which are planned to be implemented in Sprint are selected
- No features can be added during Sprint

C Burndown Chart

Artifacts

Graph representation of work done and left to be done in Sprint

- Team must have a burndown chart
- Burndown chart is a highly visible representation of work status
- Burndown chart is updated every day
- Team takes corrective actions when burndown is too high/low

Product

Artifacts

Potentially shippable product increment

- The team is expected to bring the product or system to a potentially shippable state at the end of each sprint

sam solutions

SaM Solutions Belarus,
Filimonova st., 15,
Minsk, 220037, Belarus

Tel.: +375-17-3091709
www.sam-solutions.com

Agile Project Management with Scrum
Ken Schwaber

Apply the principles of Scrum, one of the most popular agile programming methods, to software project management — and focus your team on delivering real business value.

Scrum and XP from the Trenches
Henrik Kniberg

This book aims to give you a head start by providing a detailed down-to-earth account of how one Swedish company implemented Scrum and XP with a team of approximately 40 people and how they continuously improved their process over a year's time.

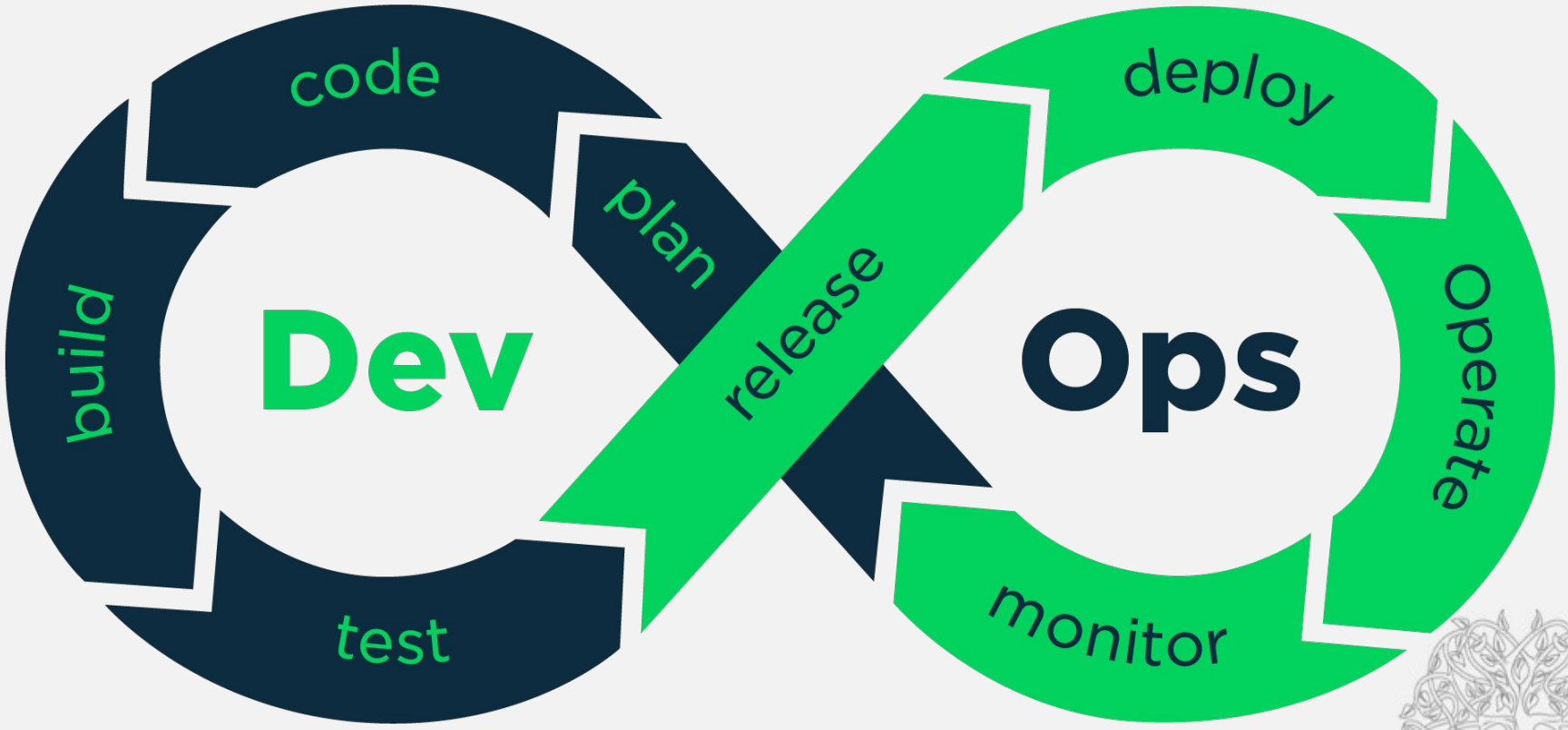
Succeeding With Agile
Mike Cohn

This is the definitive, realistic, actionable guide to starting fast with Scrum and agile–and then succeeding over the long haul.

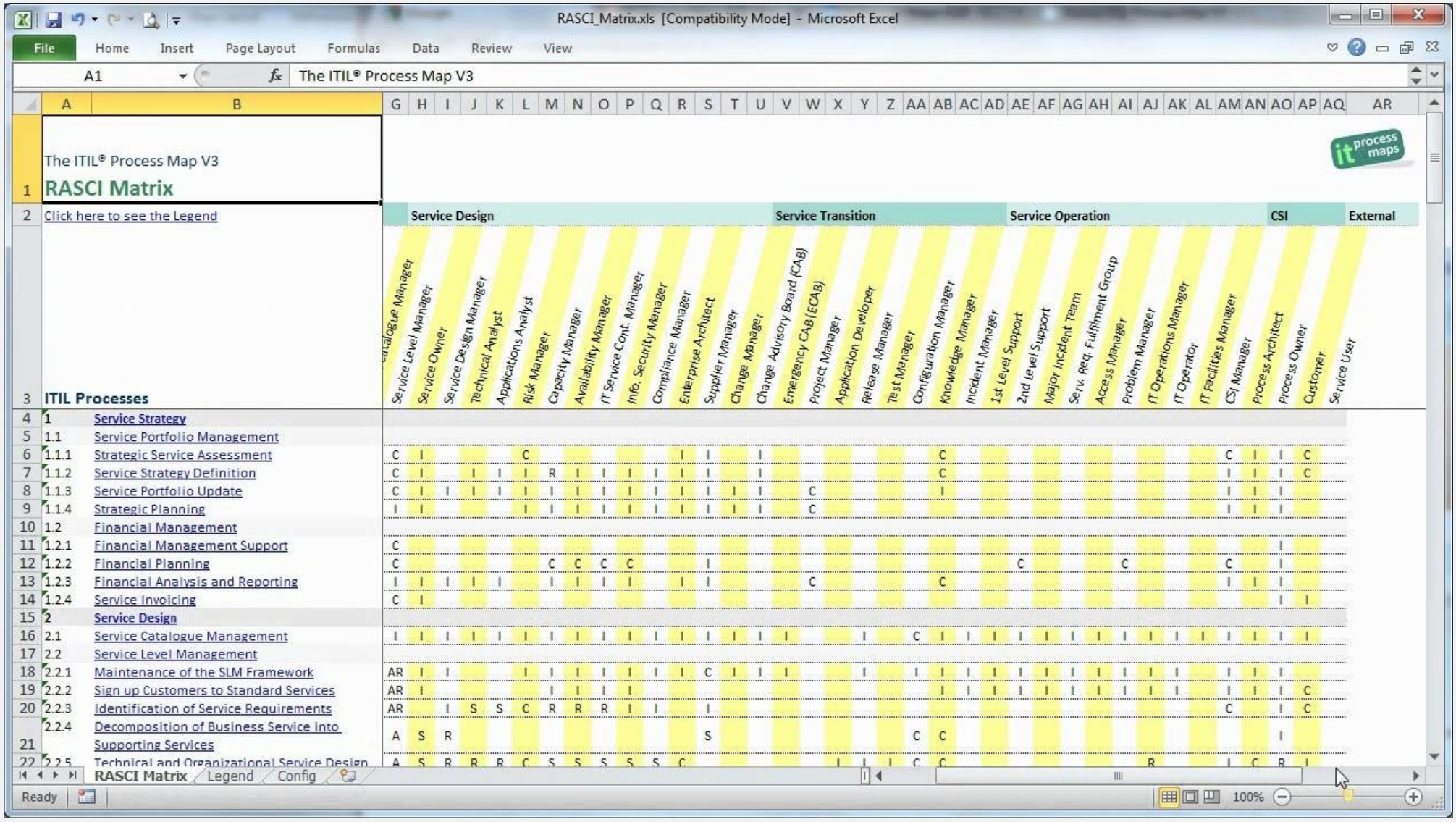
Links:

http://www.scrumalliance.org
http://www.mountaingoatsoftware.com
http://www.agilemenifesto.org
http://www.agile.by

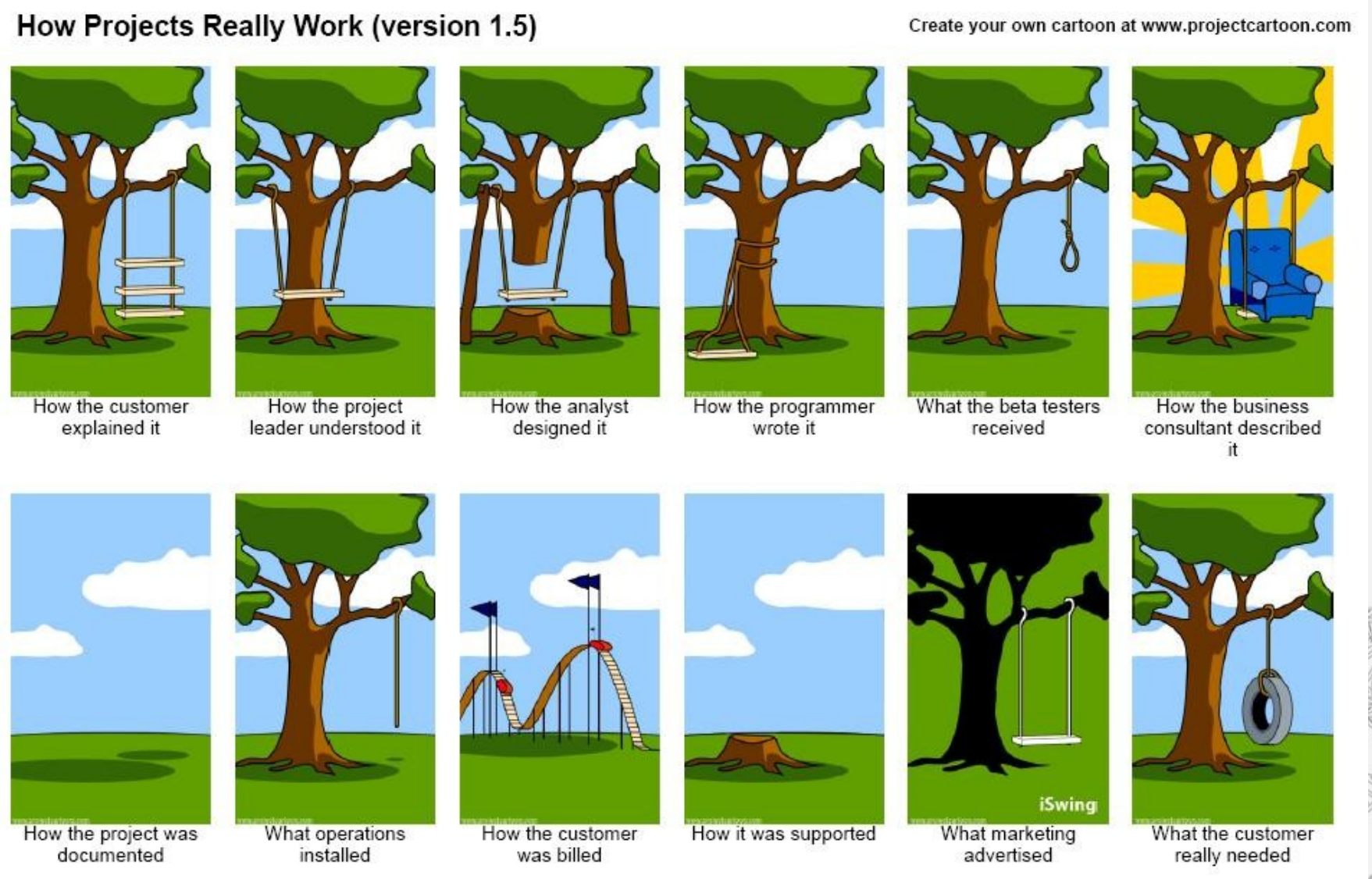
Oggi – i rilasci



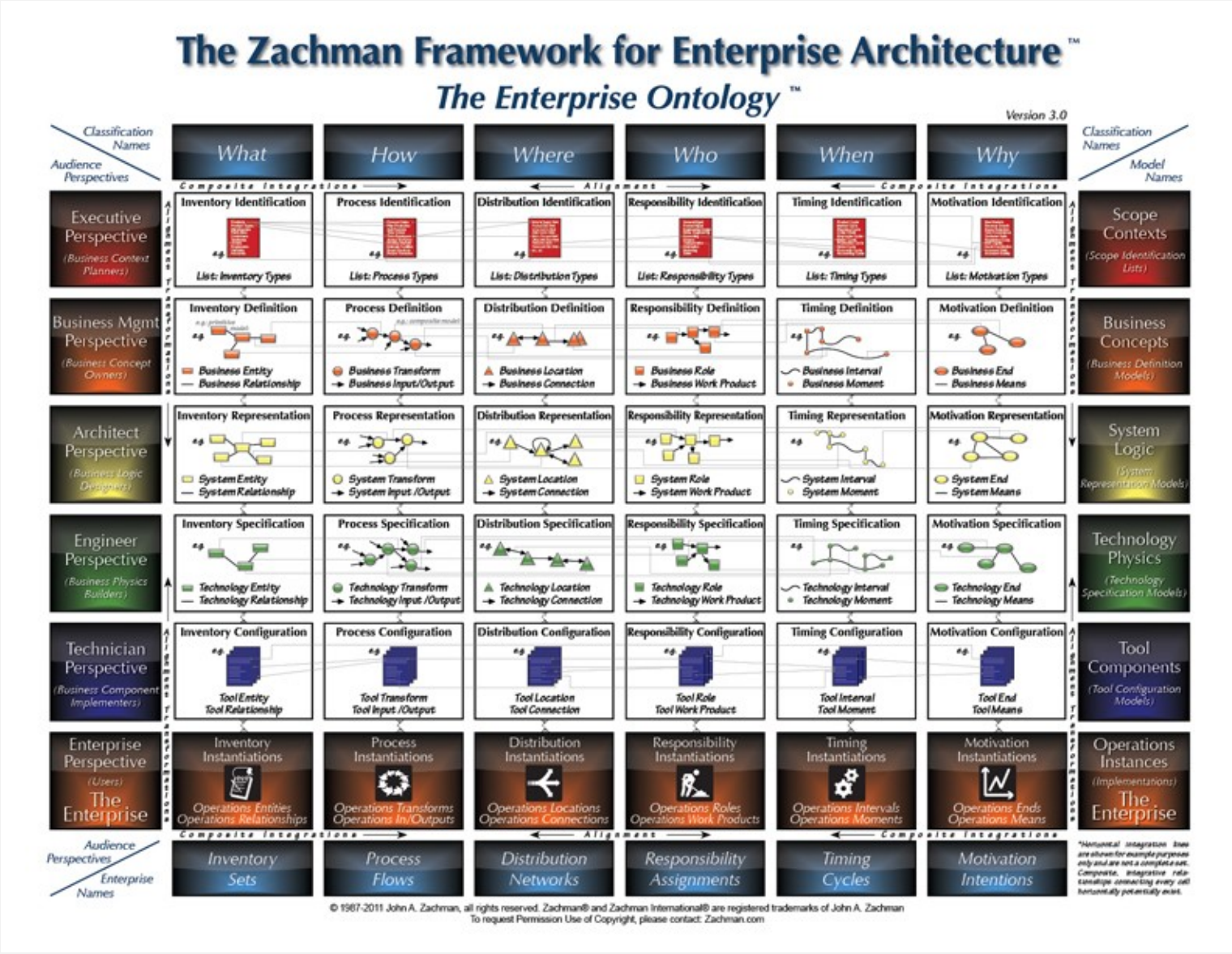
Oggi – i componenti



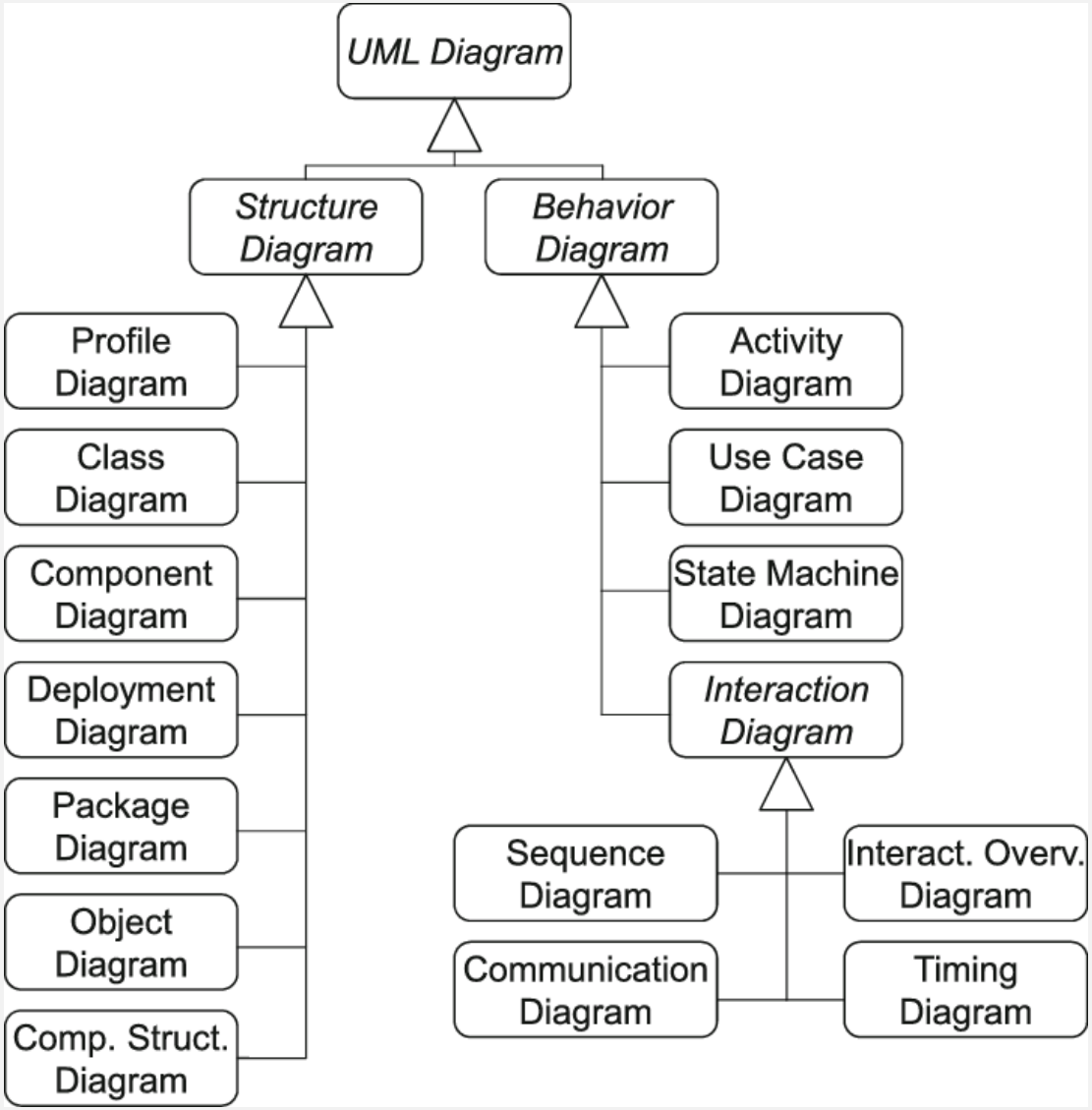
Oggi- i ruoli


How Projects Really Work (version 1.5)
Create your own cartoon at www.projectcartoon.com
How the customer explained it
How the project leader understood it
How the analyst designed it
How the programmer wrote it
What the beta testers received
How the business consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
iSwing
What marketing advertised
What the customer really needed

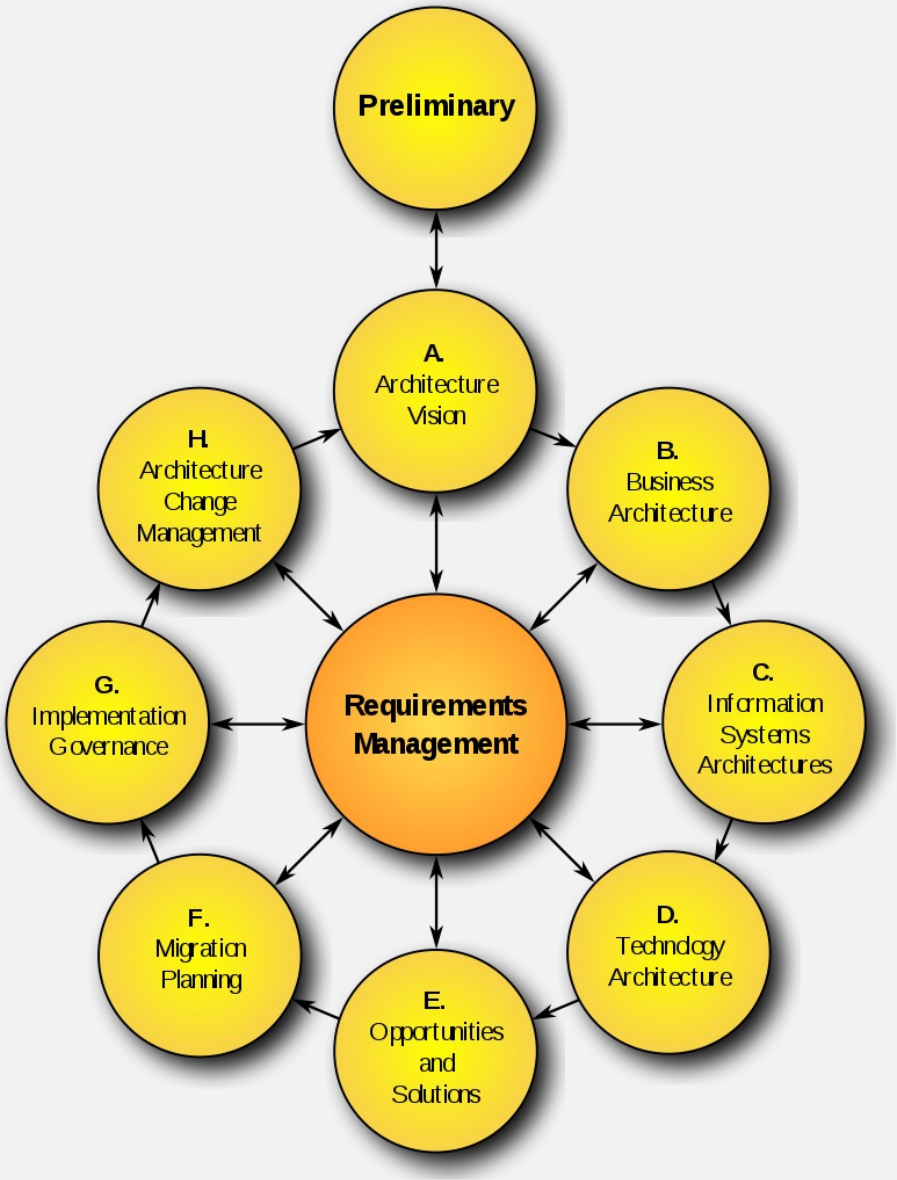
Zachman Framework®



Unified ML®



Togaf ®



Qualche volta è troppo

InformAtelier

Reazione

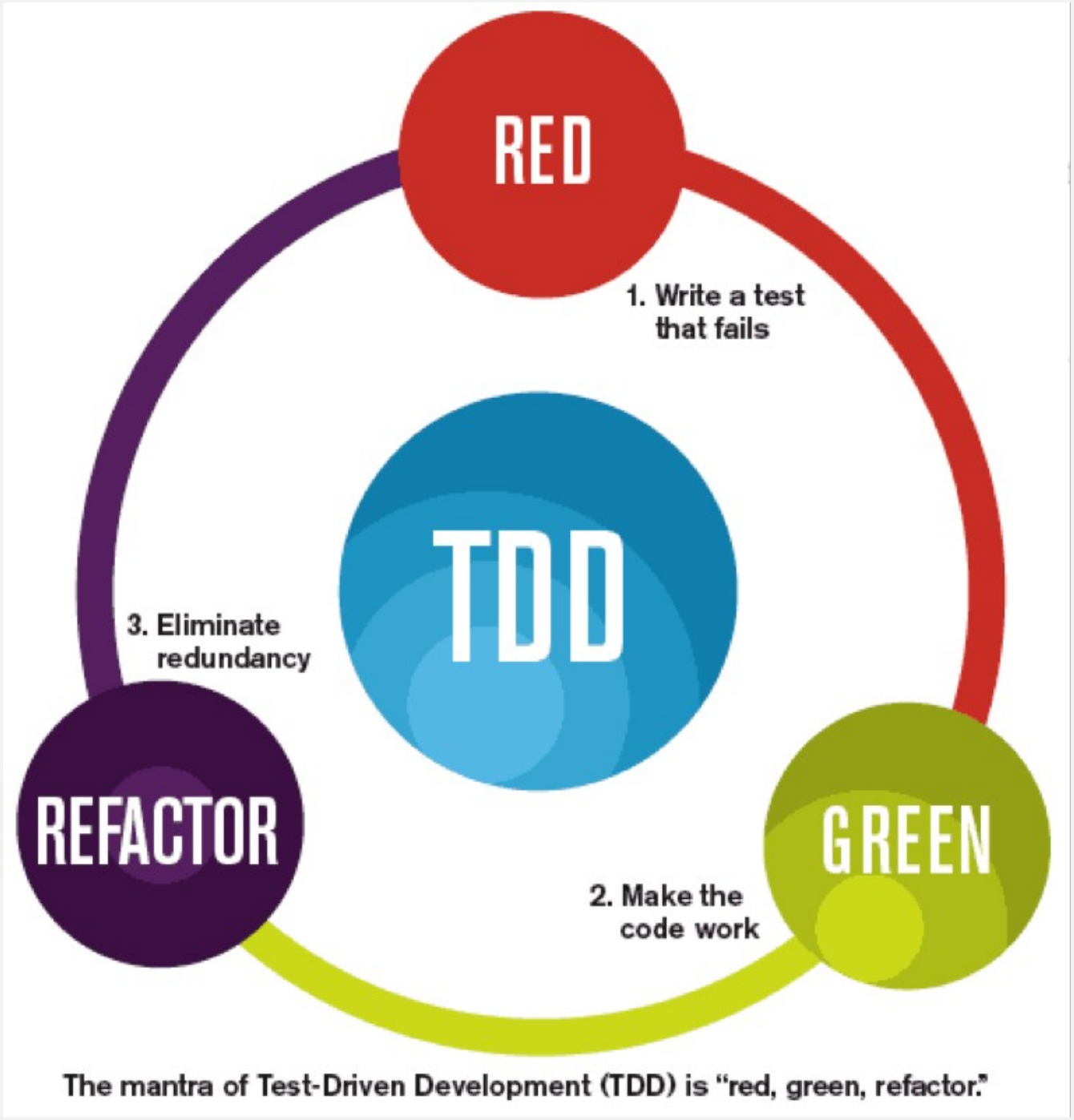

Manifesto for Agile Software Development
We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:
Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan
That is, while there is value in the items on
the right, we value the items on the left more.

Siiii , ma

quando è abbastanza esauriente?

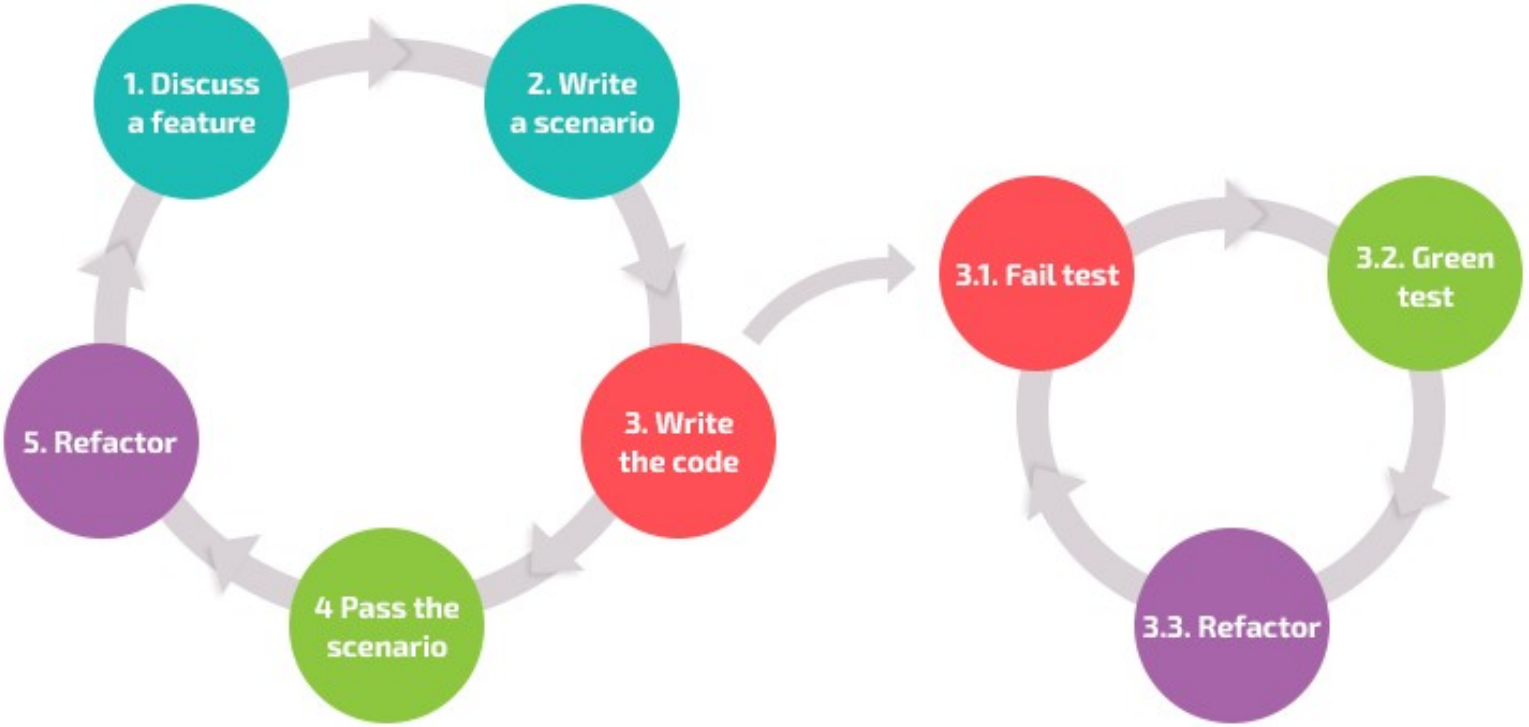
- Cosa lasciamo fuori?

InformAtelier



The mantra of Test-Driven Development (TDD) is "red, green, refactor."

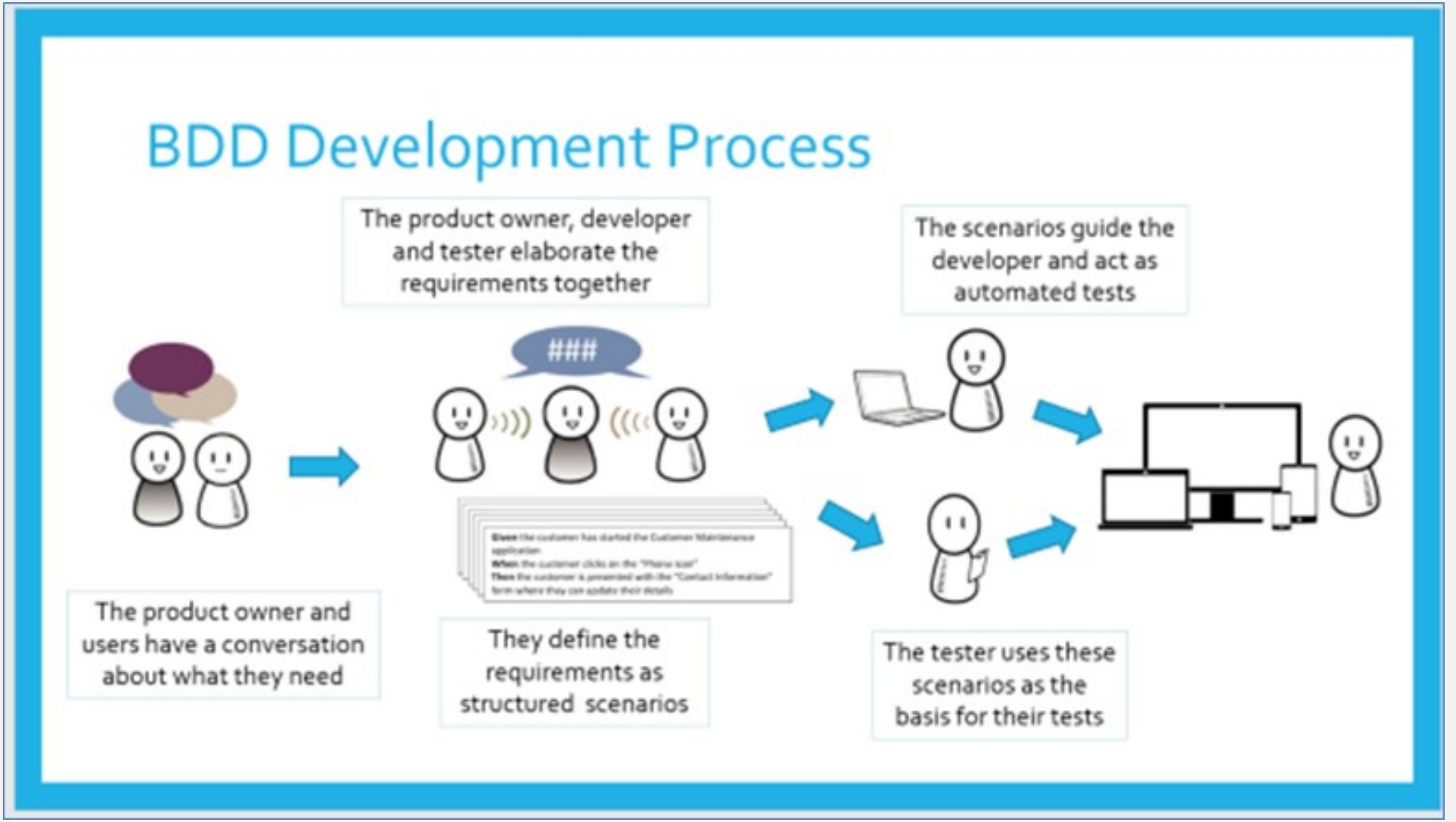
THE BDD PROCESS

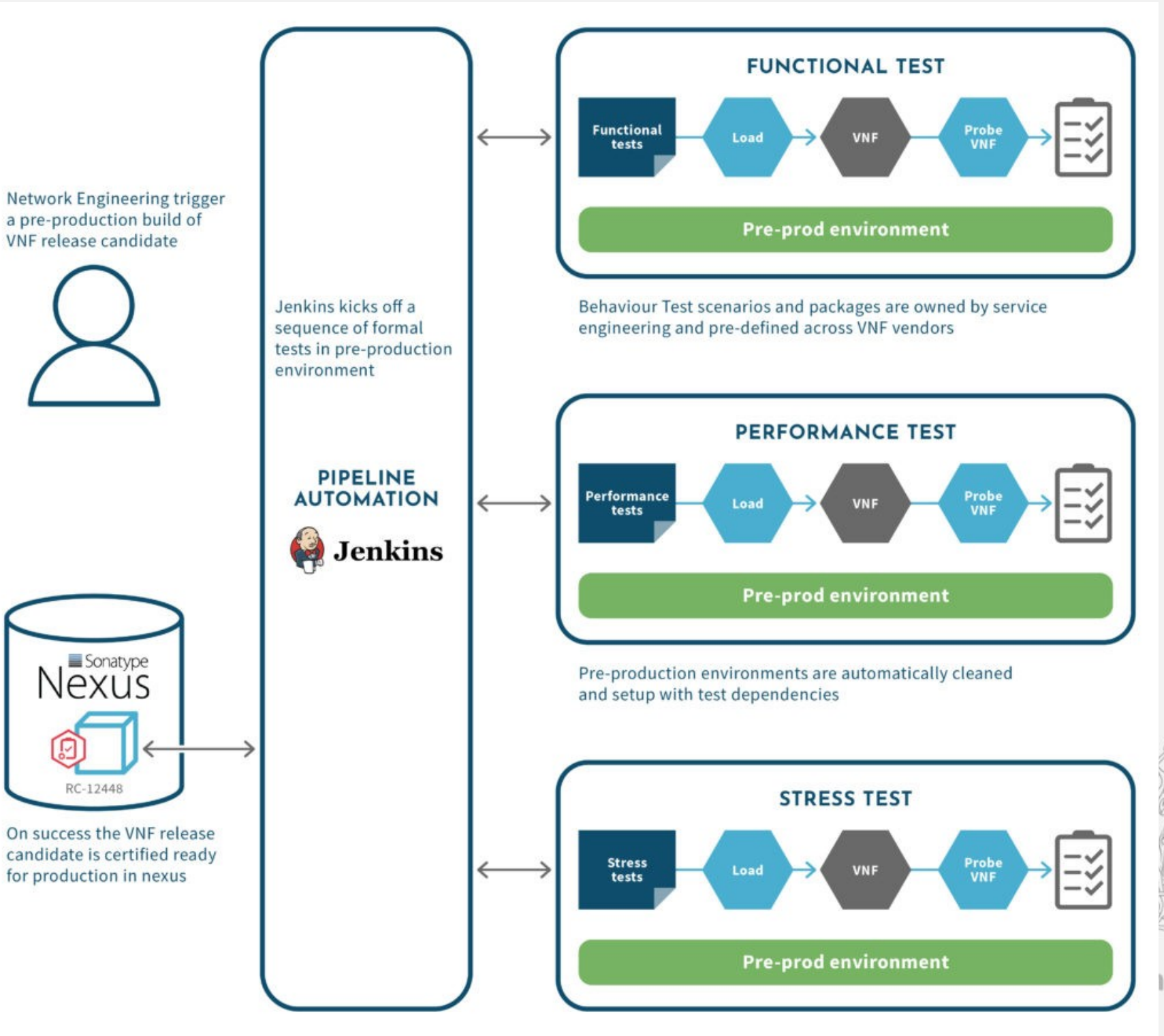


www.apriorit.com

InformAtelier

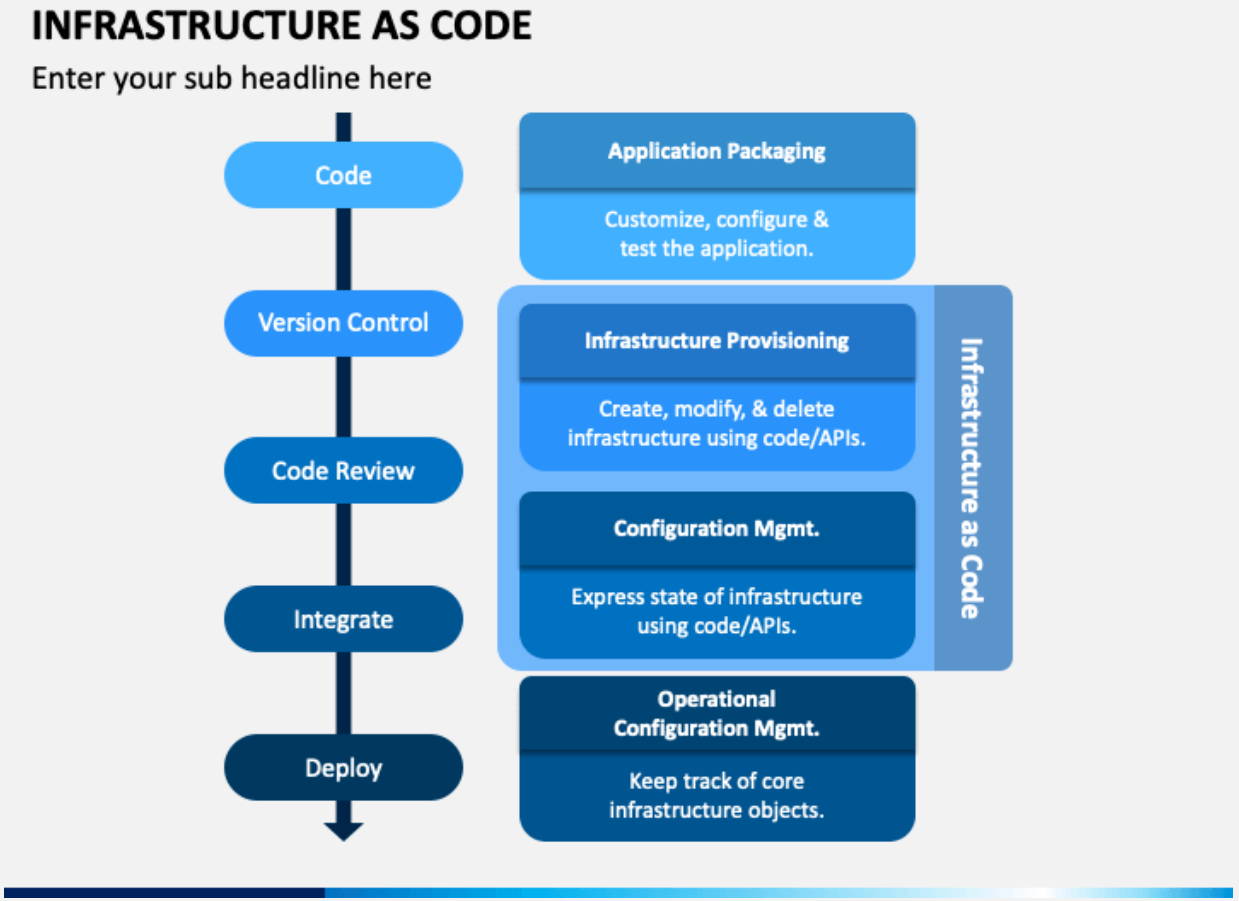
BDD Roles


BDD Development Process
The product owner, developer and tester elaborate the requirements together
The scenarios guide the developer and act as automated tests
###
The product owner and users have a conversation about what they need
They define the requirements as structured scenarios
The tester uses these scenarios as the basis for their tests

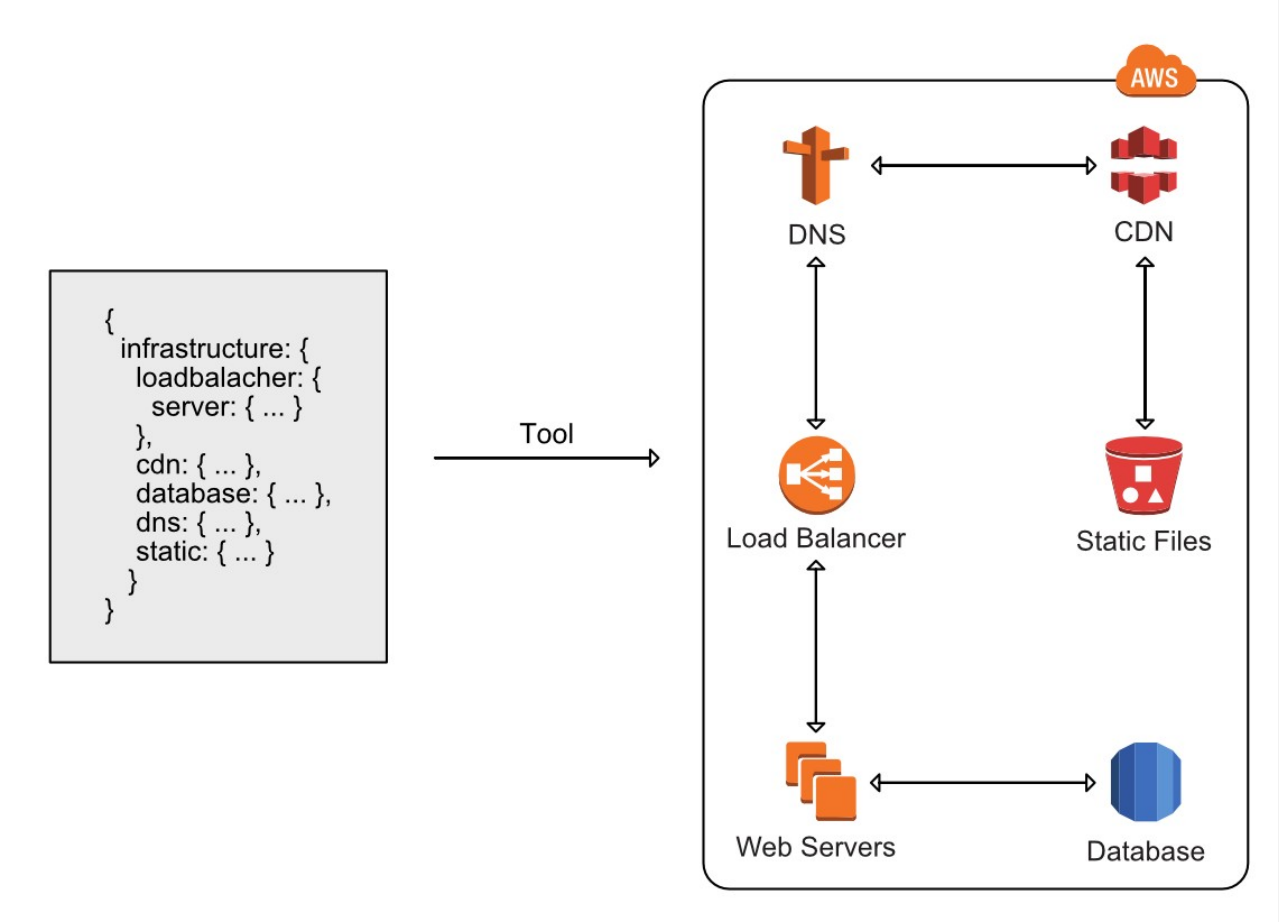

Network Engineering trigger a pre-production build of VNF release candidate
Jenkins kicks off a sequence of formal tests in pre-production environment
PIPELINE AUTOMATION
Jenkins
Sonatype Nexus
RC-12448
On success the VNF release candidate is certified ready for production in nexus
FUNCTIONAL TEST
Functional tests
Load
VNF
Probe VNF
Pre-prod environment
Behaviour Test scenarios and packages are owned by service engineering and pre-defined across VNF vendors
PERFORMANCE TEST
Performance tests
Load
VNF
Probe VNF
Pre-prod environment
Pre-production environments are automatically cleaned and setup with test dependencies
STRESS TEST
Stress tests
Load
VNF
Probe VNF
Pre-prod environment

Scegli i tuoi test

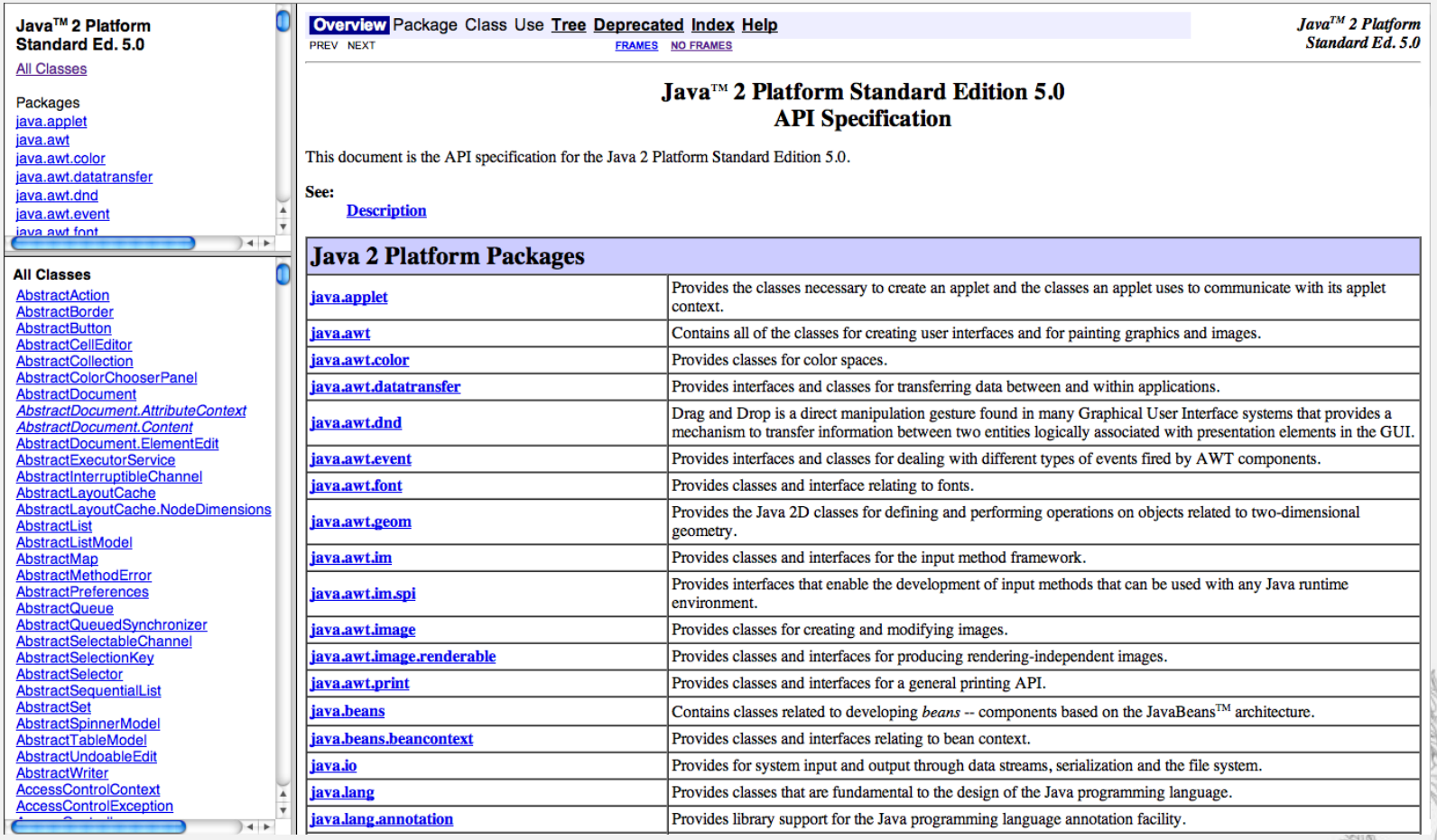
Infrastructure as Code



Infrastructure example



Doc API – JavaDoc



Java™ 2 Platform Standard Ed. 5.0

All Classes

Packages
java.applet
java.awt
java.awt.color
java.awt.datatransfer
java.awt.dnd
java.awt.event
java.awt.font

All Classes
AbstractAction
AbstractBorder
AbstractButton
AbstractCellEditor
AbstractCollection
AbstractColorChooserPanel
AbstractDocument
AbstractDocument.AttributeContext
AbstractDocument.Content
AbstractDocument.ElementEdit
AbstractExecutorService
AbstractInterruptibleChannel
AbstractLayoutCache
AbstractLayoutCache.NodeDimensions
AbstractList
AbstractListModel
AbstractMap
AbstractMethodError
AbstractPreferences
AbstractQueue
AbstractQueuedSynchronizer
AbstractSelectableChannel
AbstractSelectionKey
AbstractSelector
AbstractSequentialList
AbstractSet
AbstractSpinnerModel
AbstractTableModel
AbstractUndoableEdit
AbstractWriter
AccessControlContext
AccessControlException

Overview Package Class Use **Tree** **Deprecated** **Index** **Help**
PREV NEXT FRAMES NO FRAMES

Java™ 2 Platform Standard Ed. 5.0

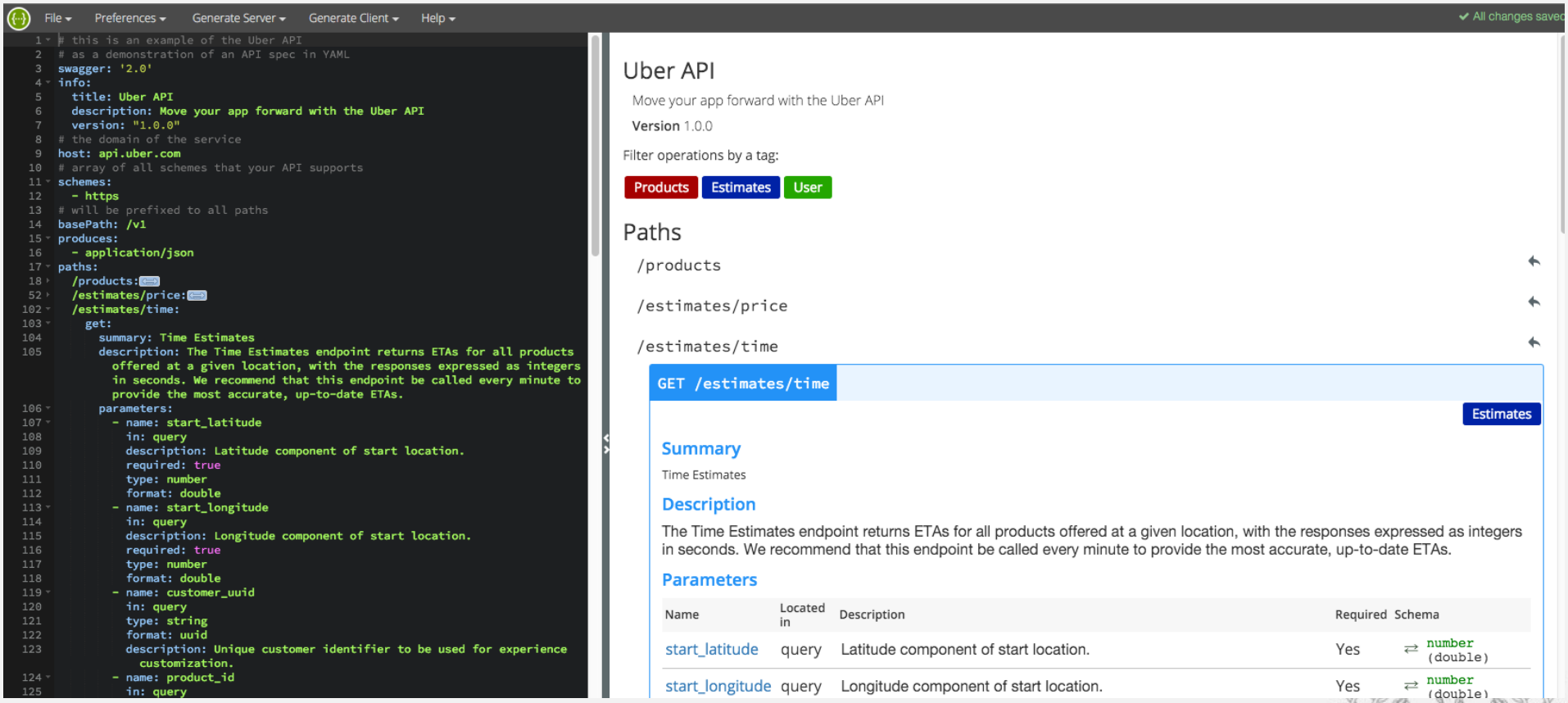
Java™ 2 Platform Standard Edition 5.0
API Specification

This document is the API specification for the Java 2 Platform Standard Edition 5.0.

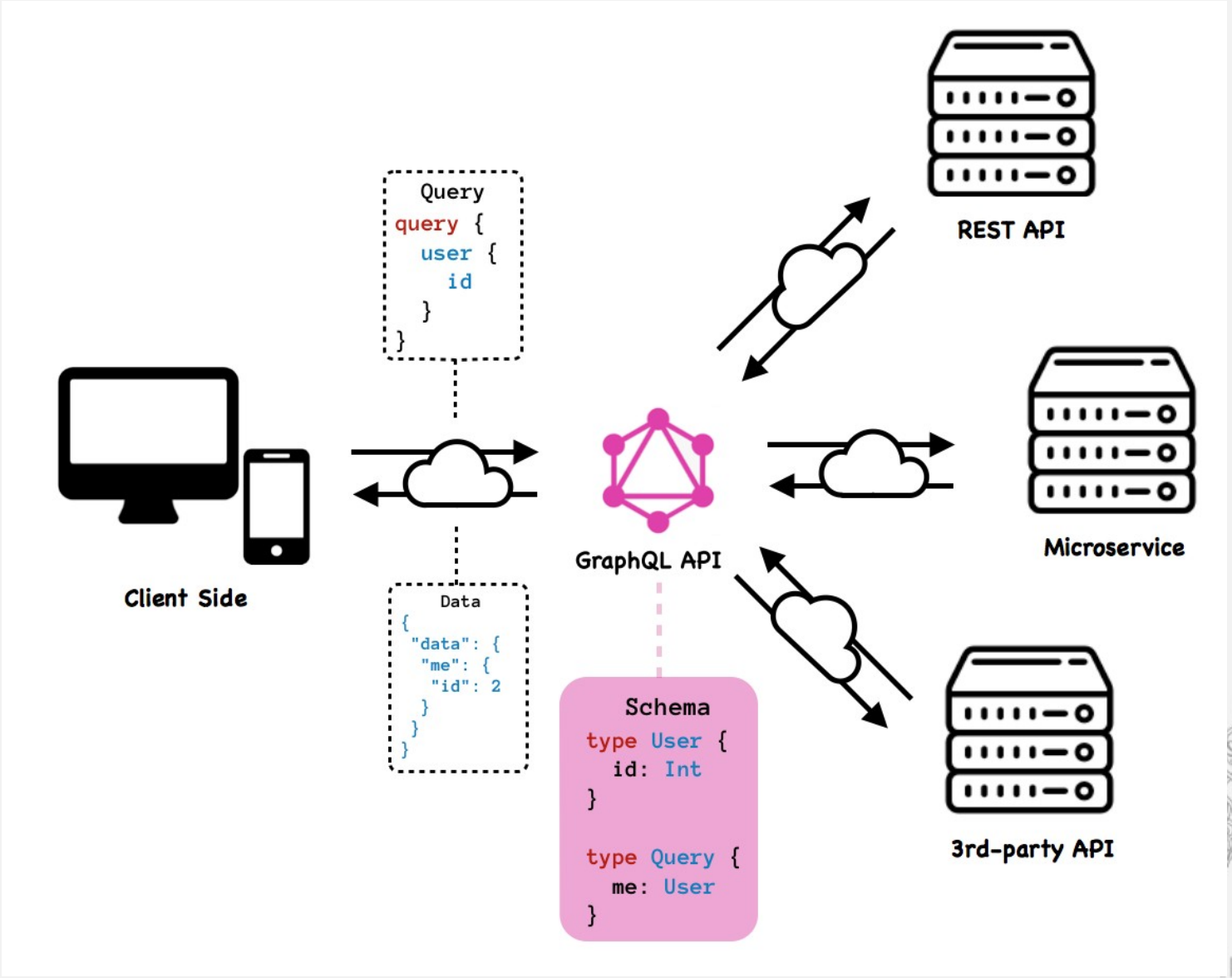
See:
Description

Java 2 Platform Packages	
java.applet	Provides the classes necessary to create an applet and the classes an applet uses to communicate with its applet context.
java.awt	Contains all of the classes for creating user interfaces and for painting graphics and images.
java.awt.color	Provides classes for color spaces.
java.awt.datatransfer	Provides interfaces and classes for transferring data between and within applications.
java.awt.dnd	Drag and Drop is a direct manipulation gesture found in many Graphical User Interface systems that provides a mechanism to transfer information between two entities logically associated with presentation elements in the GUI.
java.awt.event	Provides interfaces and classes for dealing with different types of events fired by AWT components.
java.awt.font	Provides classes and interface relating to fonts.
java.awt.geom	Provides the Java 2D classes for defining and performing operations on objects related to two-dimensional geometry.
java.awt.im	Provides classes and interfaces for the input method framework.
java.awt.im.spi	Provides interfaces that enable the development of input methods that can be used with any Java runtime environment.
java.awt.image	Provides classes for creating and modifying images.
java.awt.image.renderable	Provides classes and interfaces for producing rendering-independent images.
java.awt.print	Provides classes and interfaces for a general printing API.
java.beans	Contains classes related to developing *beans* -- components based on the JavaBeans™ architecture.
java.beans.beancontext	Provides classes and interfaces relating to bean context.
java.io	Provides for system input and output through data streams, serialization and the file system.
java.lang	Provides classes that are fundamental to the design of the Java programming language.
java.lang.annotation	Provides library support for the Java programming language annotation facility.

Doc API - OpenAPI

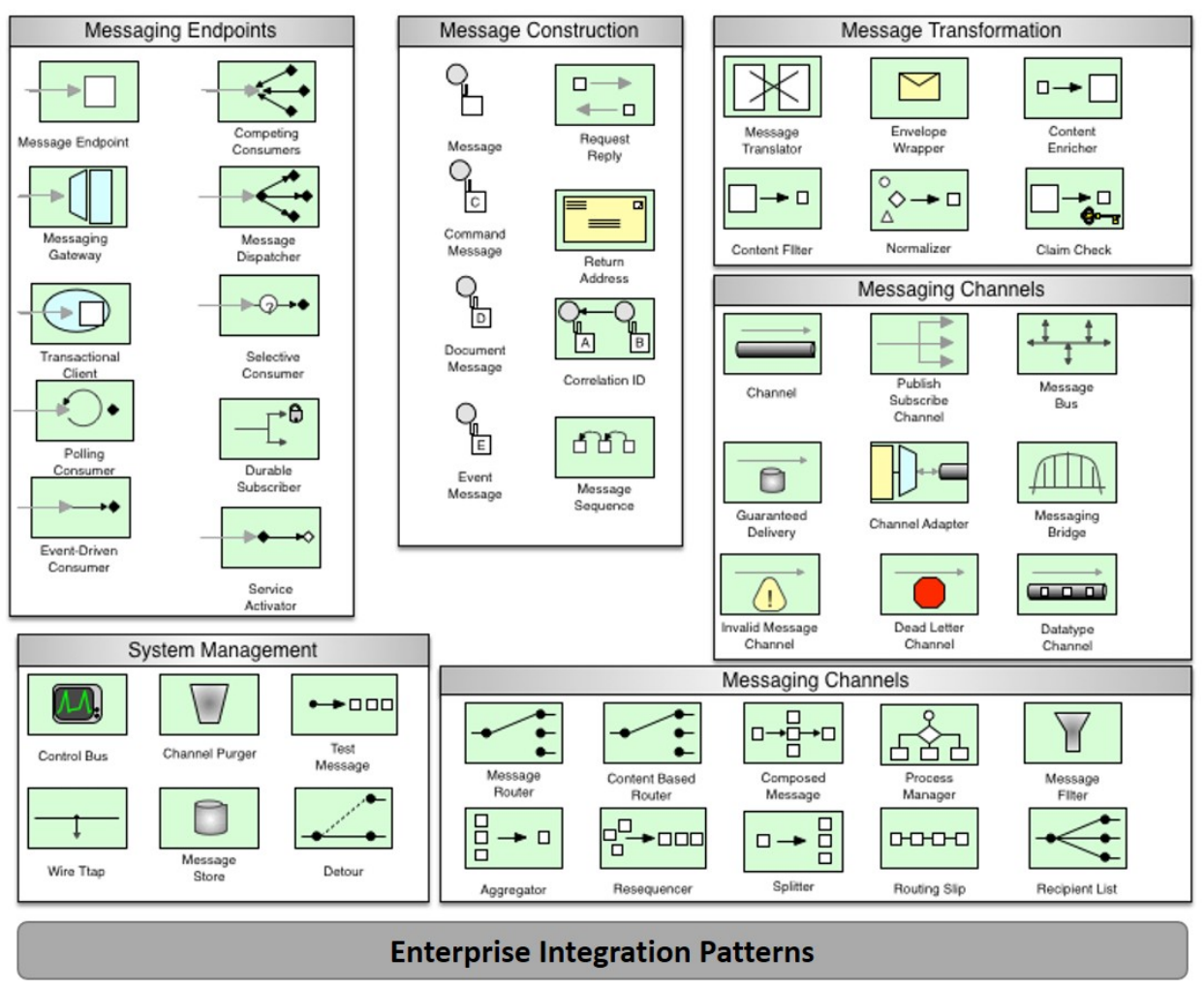


Doc API - GraphQL

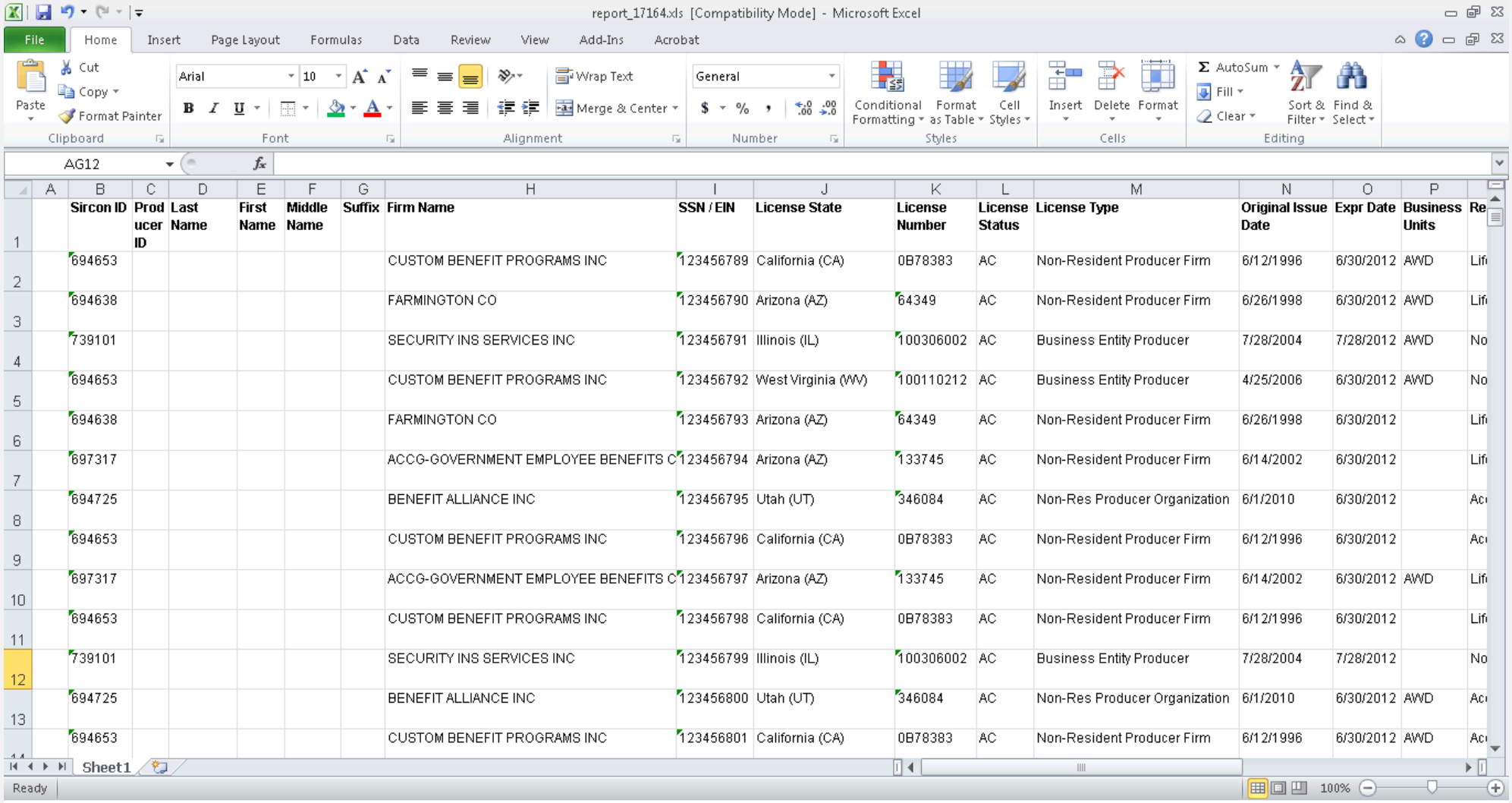


Framework Patterns

- Conventions & Vocabulary

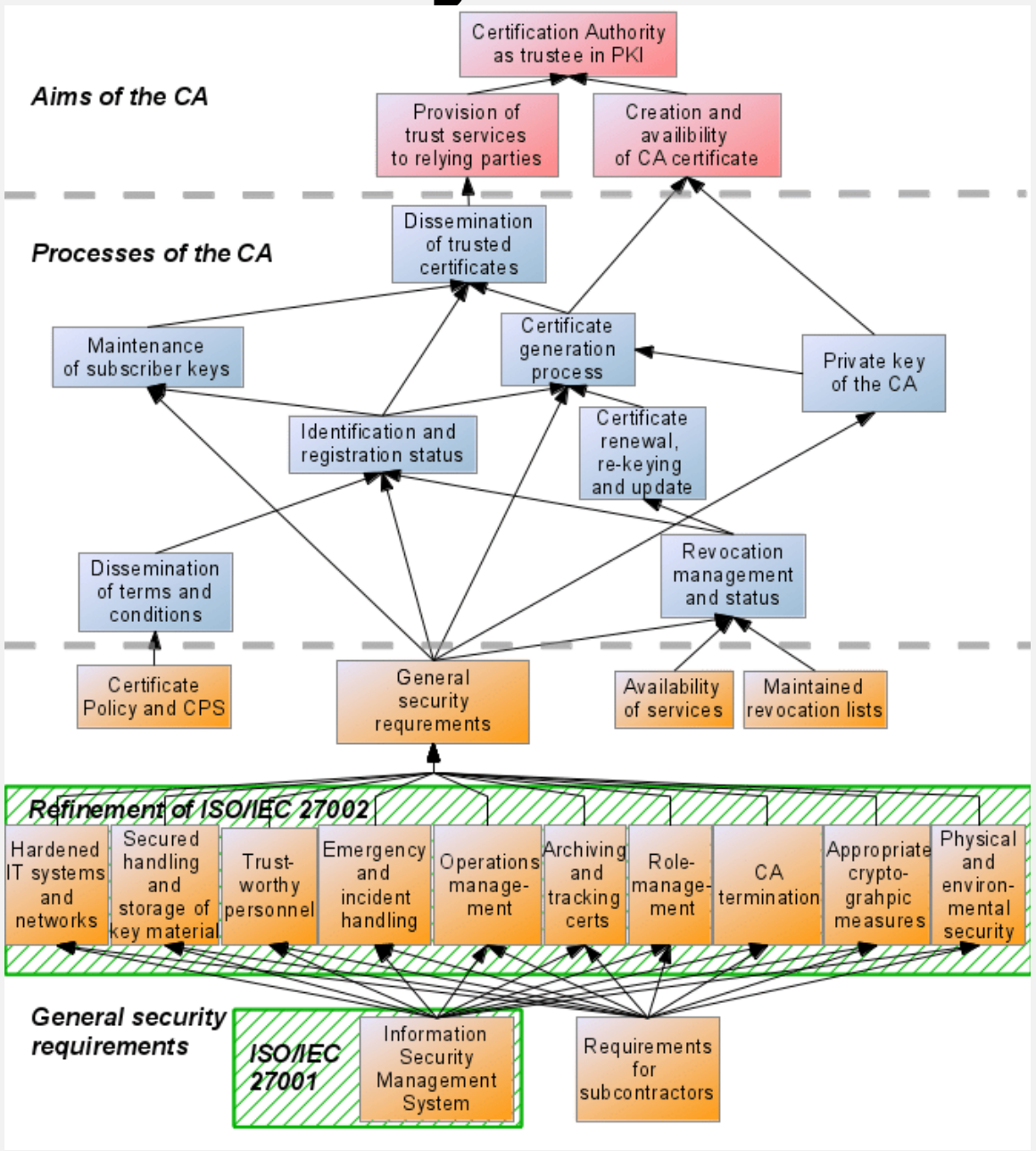


Licences Doc

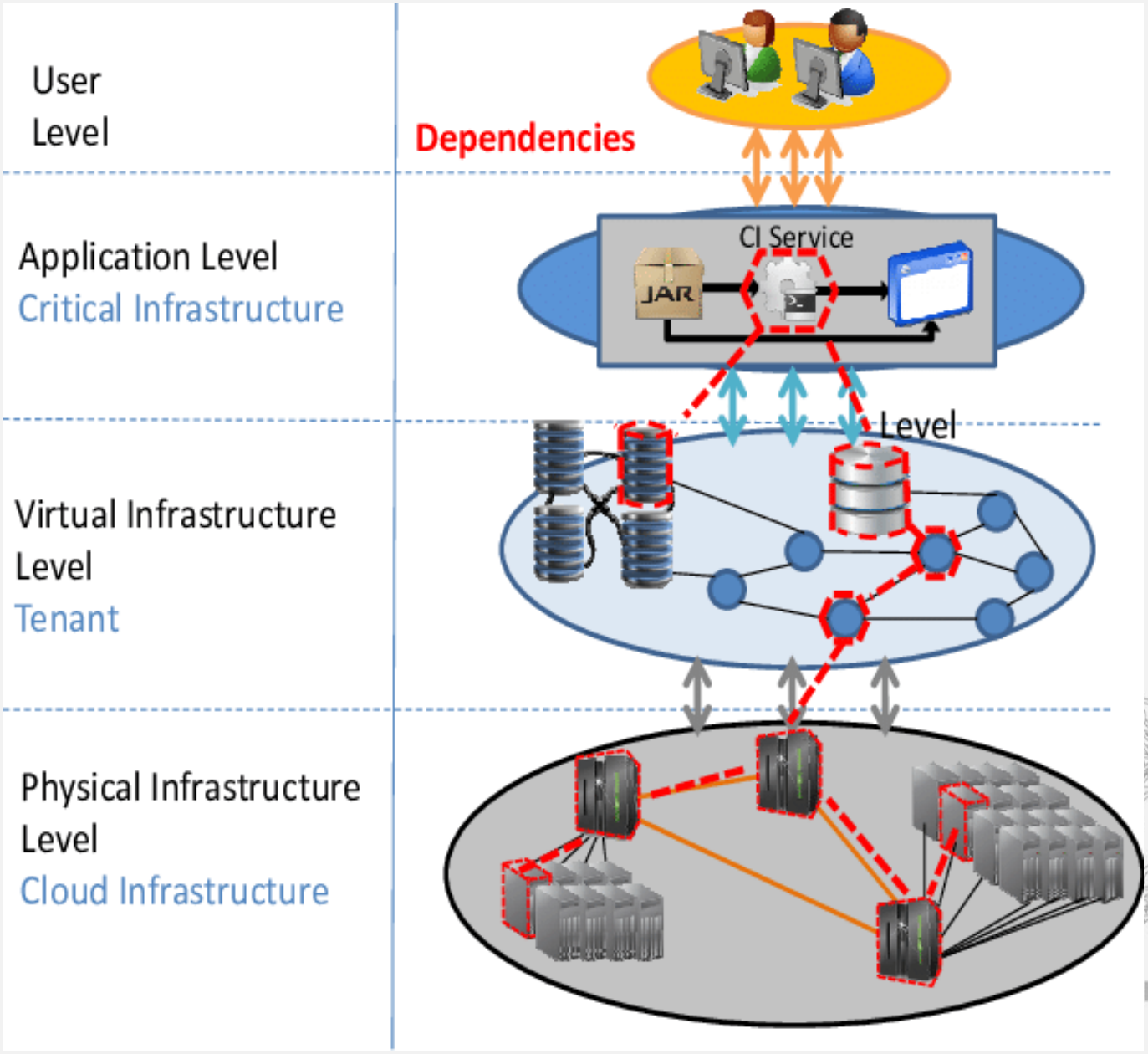


	Sircon ID	Producer ID	Last Name	First Name	Middle Name	Suffix	Firm Name	SSN / EIN	License State	License Number	License Status	License Type	Original Issue Date	Expr Date	Business Units	Re
2	694653						CUSTOM BENEFIT PROGRAMS INC	123456789	California (CA)	0B78383	AC	Non-Resident Producer Firm	6/12/1996	6/30/2012	AWD	Lif
3	694638						FARMINGTON CO	123456790	Arizona (AZ)	64349	AC	Non-Resident Producer Firm	6/26/1998	6/30/2012	AWD	Lif
4	739101						SECURITY INS SERVICES INC	123456791	Illinois (IL)	100306002	AC	Business Entity Producer	7/28/2004	7/28/2012	AWD	No
5	694653						CUSTOM BENEFIT PROGRAMS INC	123456792	West Virginia (WV)	100110212	AC	Business Entity Producer	4/25/2006	6/30/2012	AWD	No
6	694638						FARMINGTON CO	123456793	Arizona (AZ)	64349	AC	Non-Resident Producer Firm	6/26/1998	6/30/2012		Lif
7	697317						ACCG-GOVERNMENT EMPLOYEE BENEFITS C	123456794	Arizona (AZ)	133745	AC	Non-Resident Producer Firm	6/14/2002	6/30/2012		Lif
8	694725						BENEFIT ALLIANCE INC	123456795	Utah (UT)	346084	AC	Non-Res Producer Organization	6/1/2010	6/30/2012		Ac
9	694653						CUSTOM BENEFIT PROGRAMS INC	123456796	California (CA)	0B78383	AC	Non-Resident Producer Firm	6/12/1996	6/30/2012		Ac
10	697317						ACCG-GOVERNMENT EMPLOYEE BENEFITS C	123456797	Arizona (AZ)	133745	AC	Non-Resident Producer Firm	6/14/2002	6/30/2012	AWD	Lif
11	694653						CUSTOM BENEFIT PROGRAMS INC	123456798	California (CA)	0B78383	AC	Non-Resident Producer Firm	6/12/1996	6/30/2012		Lif
12	739101						SECURITY INS SERVICES INC	123456799	Illinois (IL)	100306002	AC	Business Entity Producer	7/28/2004	7/28/2012		No
13	694725						BENEFIT ALLIANCE INC	123456800	Utah (UT)	346084	AC	Non-Res Producer Organization	6/1/2010	6/30/2012	AWD	Ac
	694653						CUSTOM BENEFIT PROGRAMS INC	123456801	California (CA)	0B78383	AC	Non-Resident Producer Firm	6/12/1996	6/30/2012	AWD	Ac

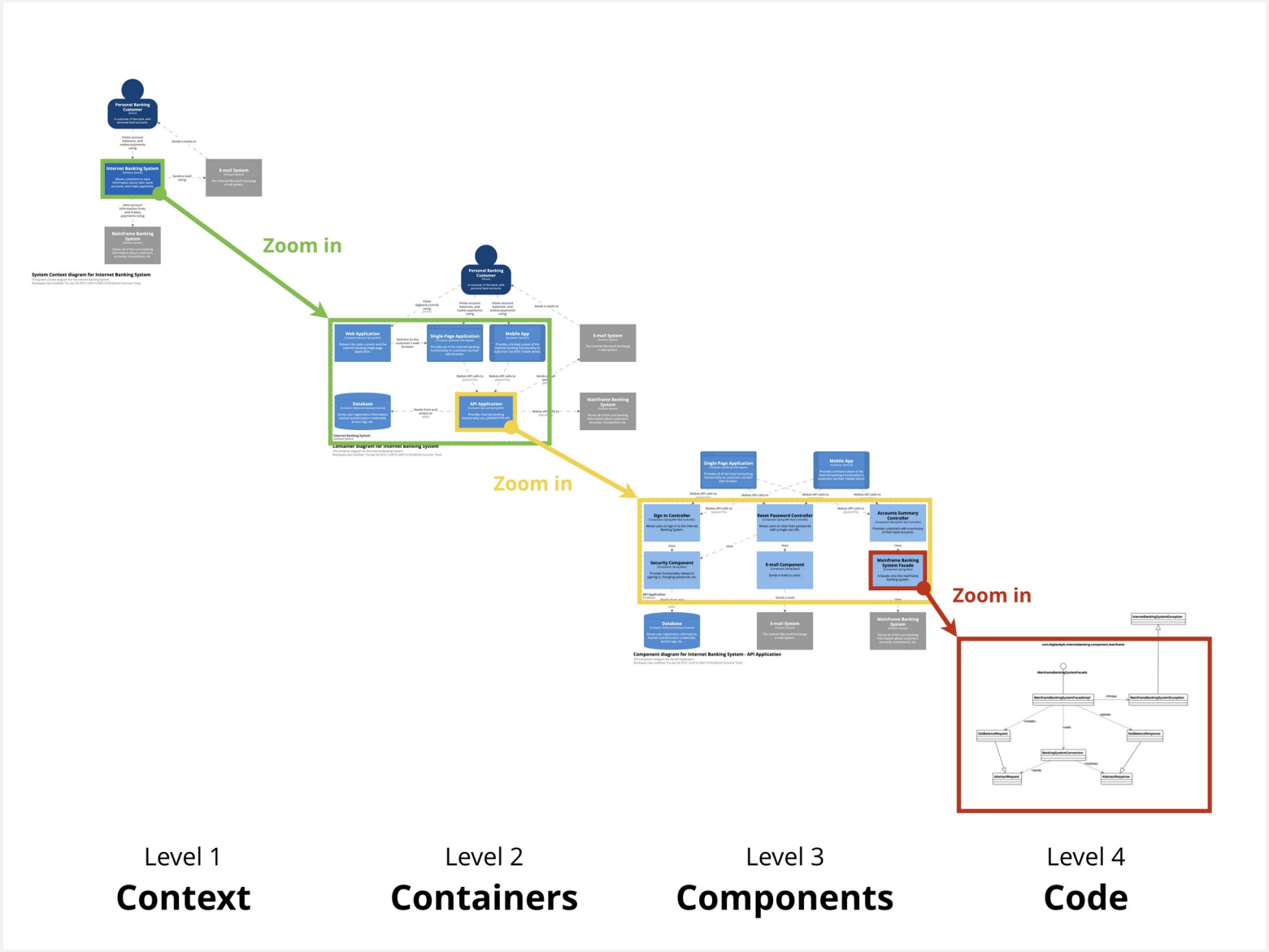
Security Chains



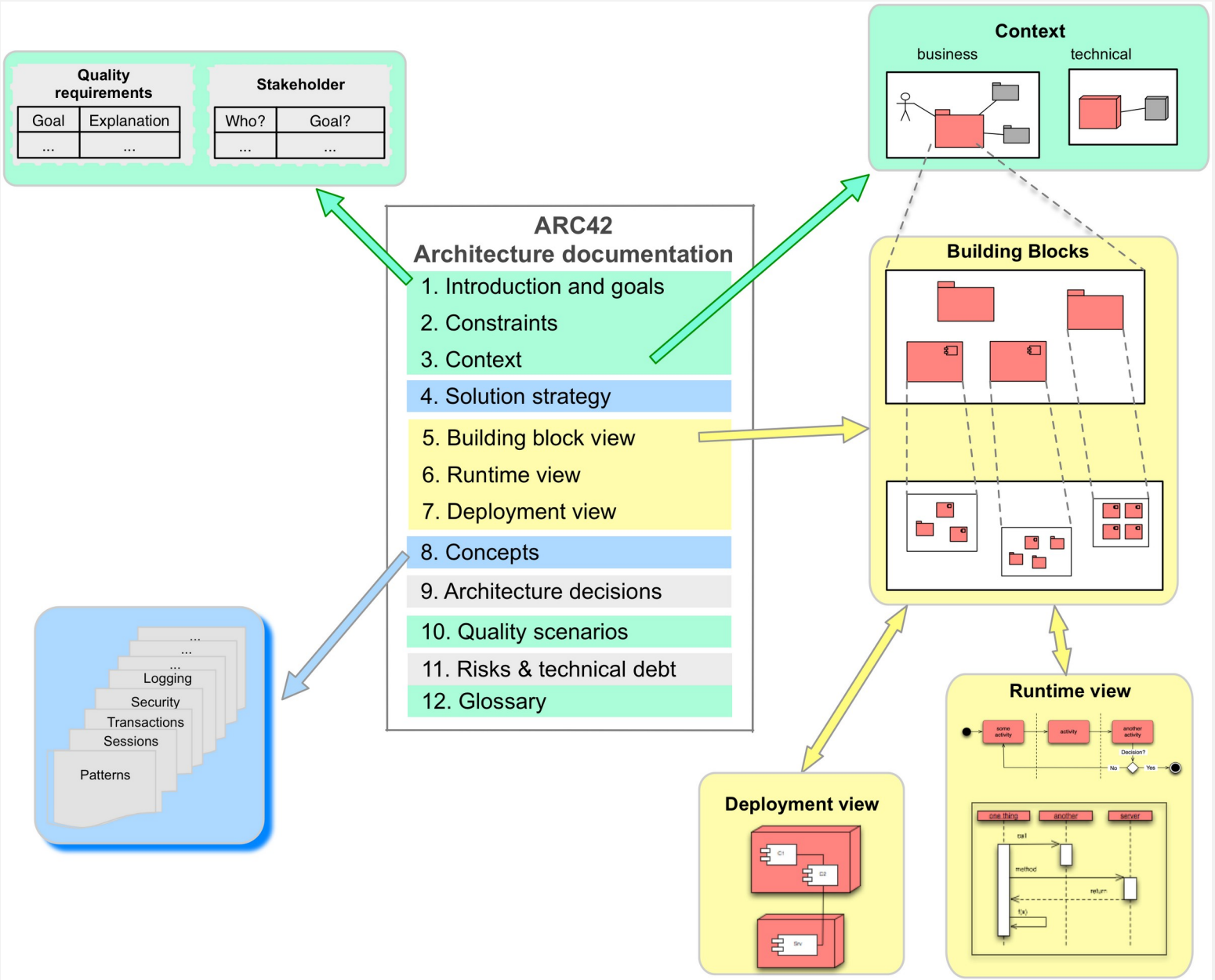
Security Levels



c4model



arc42



InformAtelier

Architecture Decision Record

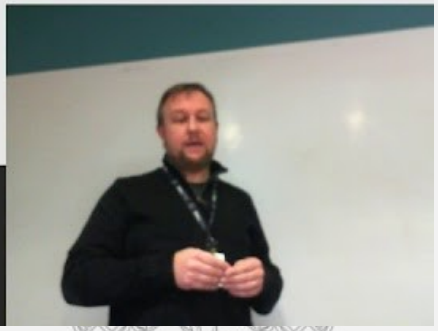
Title – short noun phrase e.g. “Use Kotlin for BackEnd Microservice”

Status – Proposed/Accepted/Deprecated/Superseded

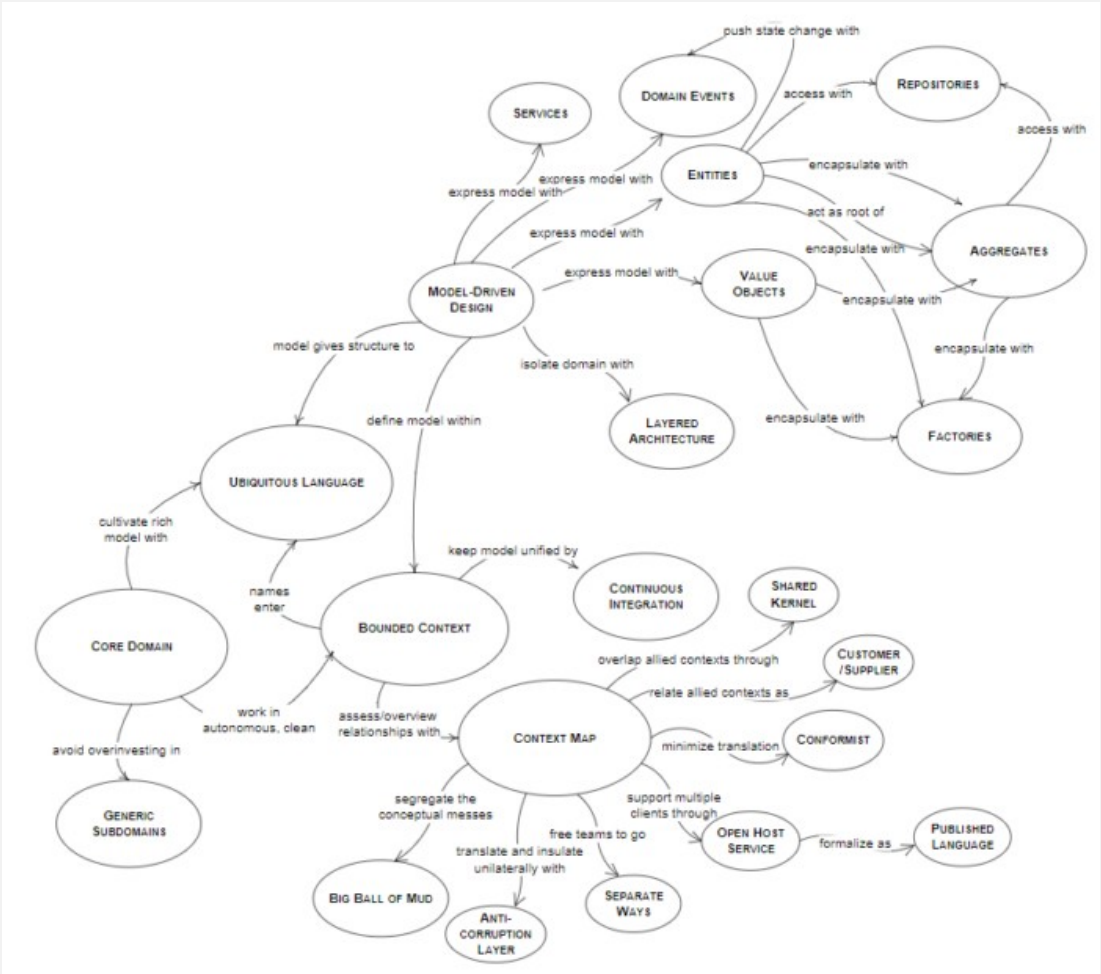
Context – forces at play…often in tension & value neutral (just the facts)

Decision - "We will ...“

Consequences - All consequences positive, neutral and negative

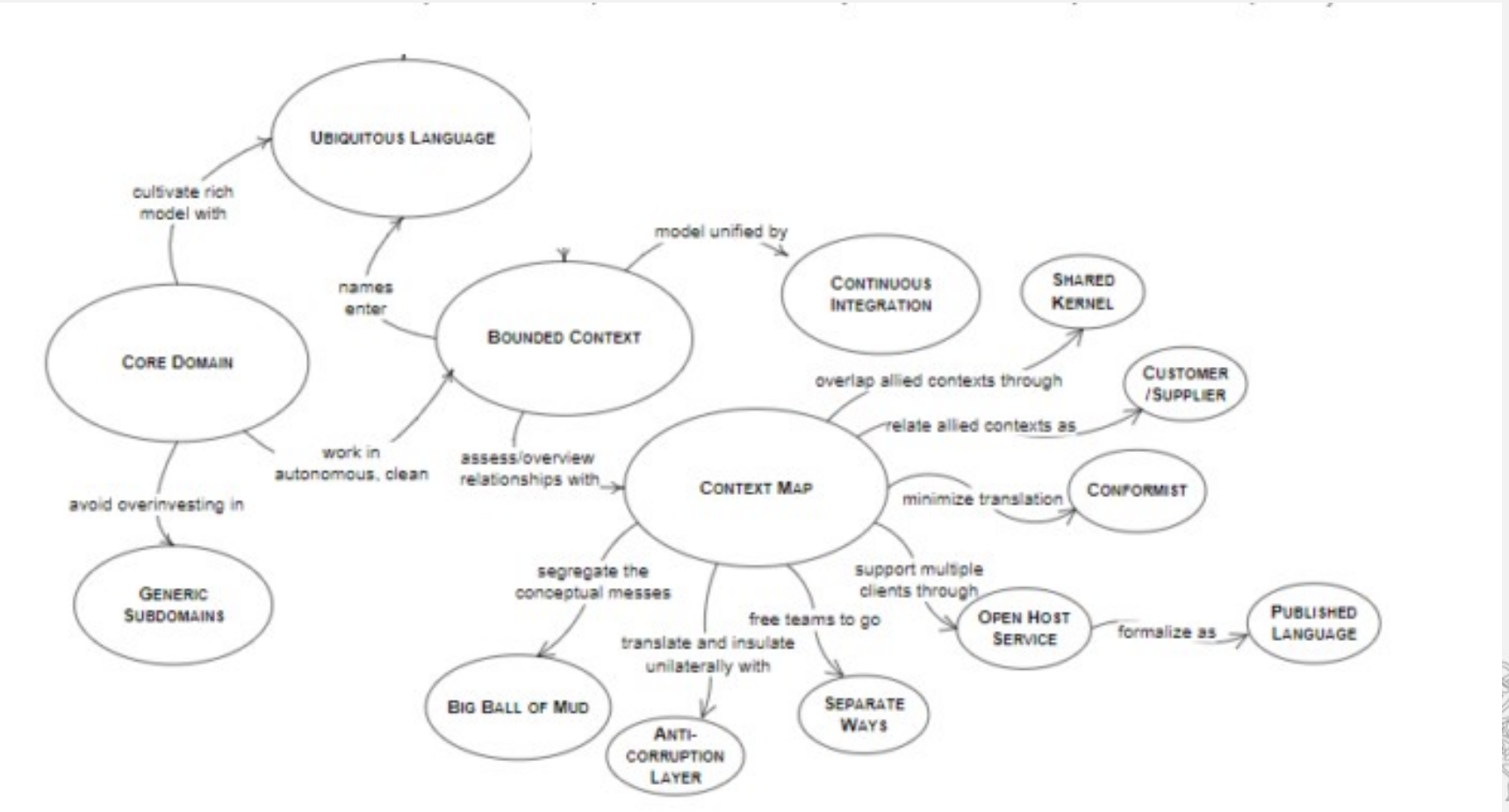


Domain Driven Design

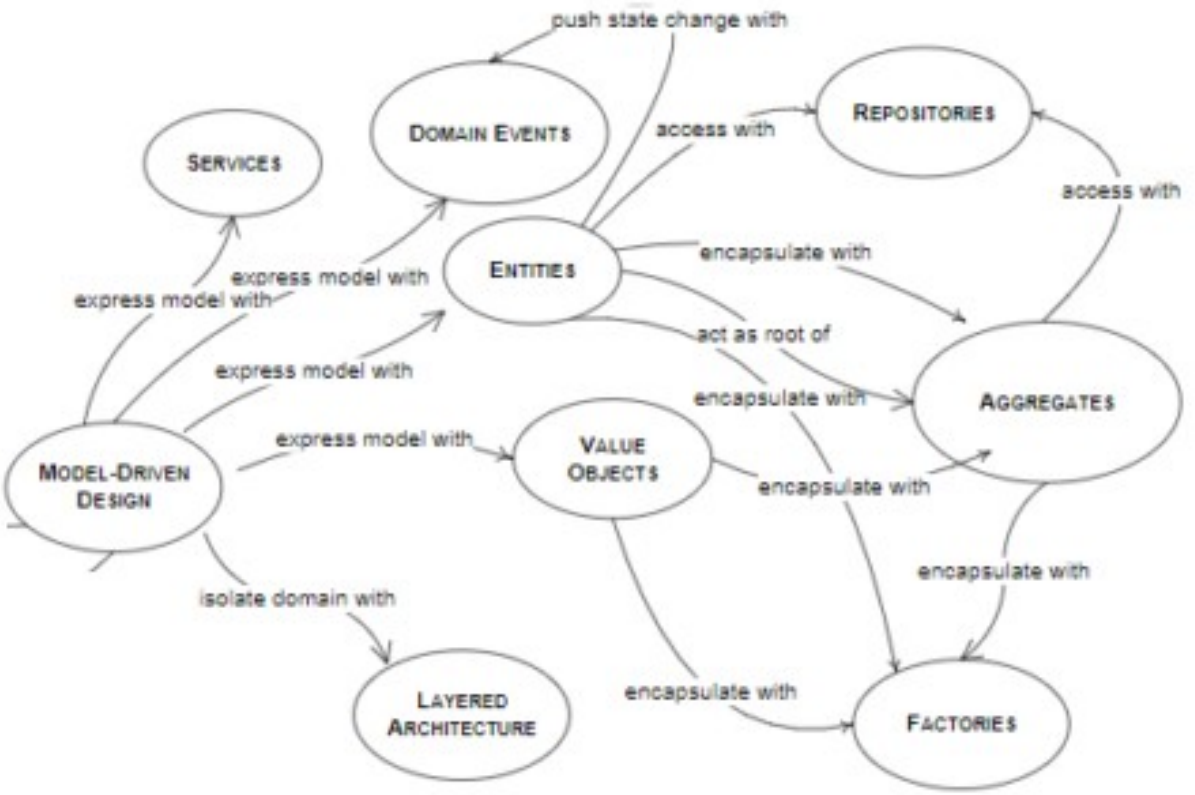


InformAtelier

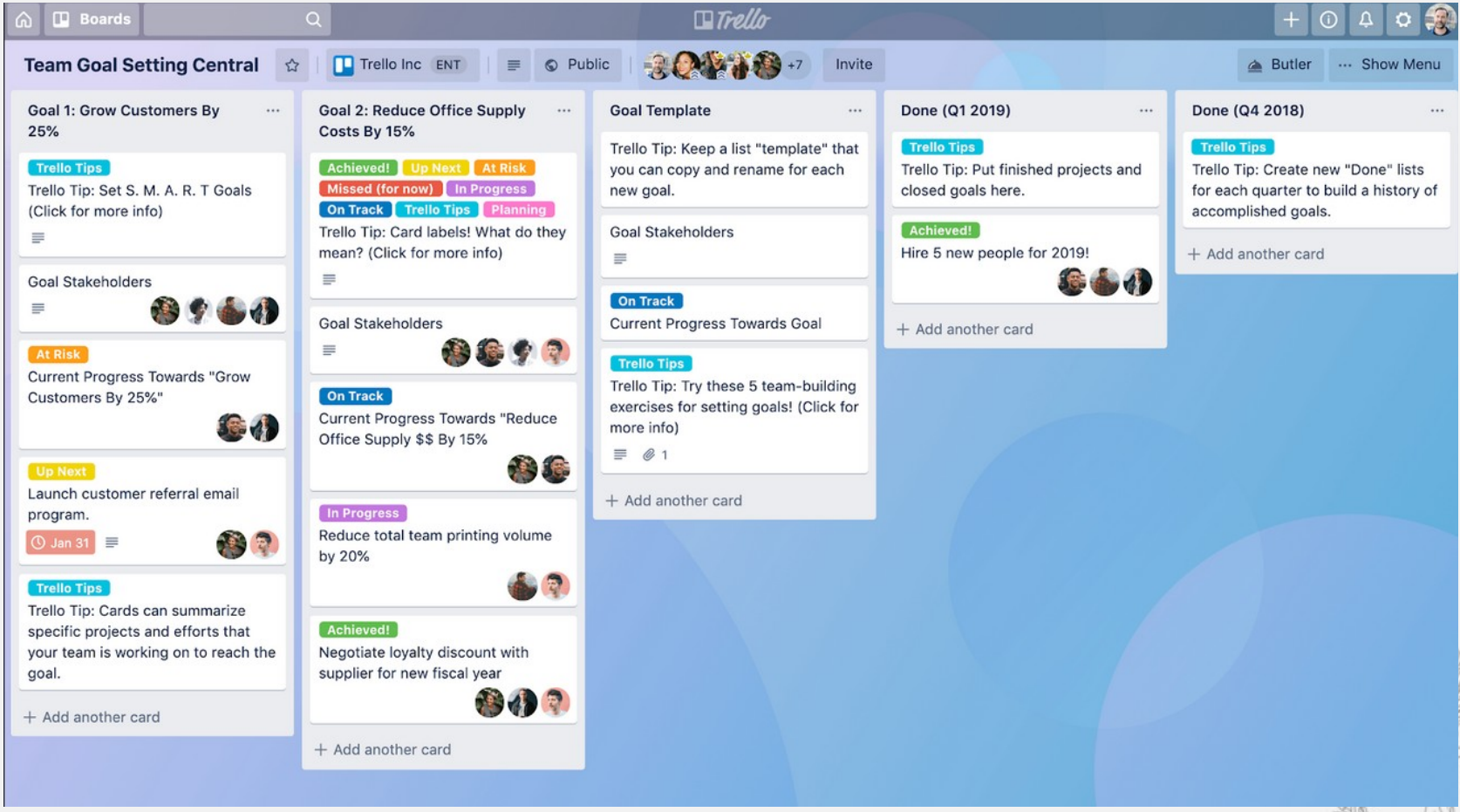
DDD strategic



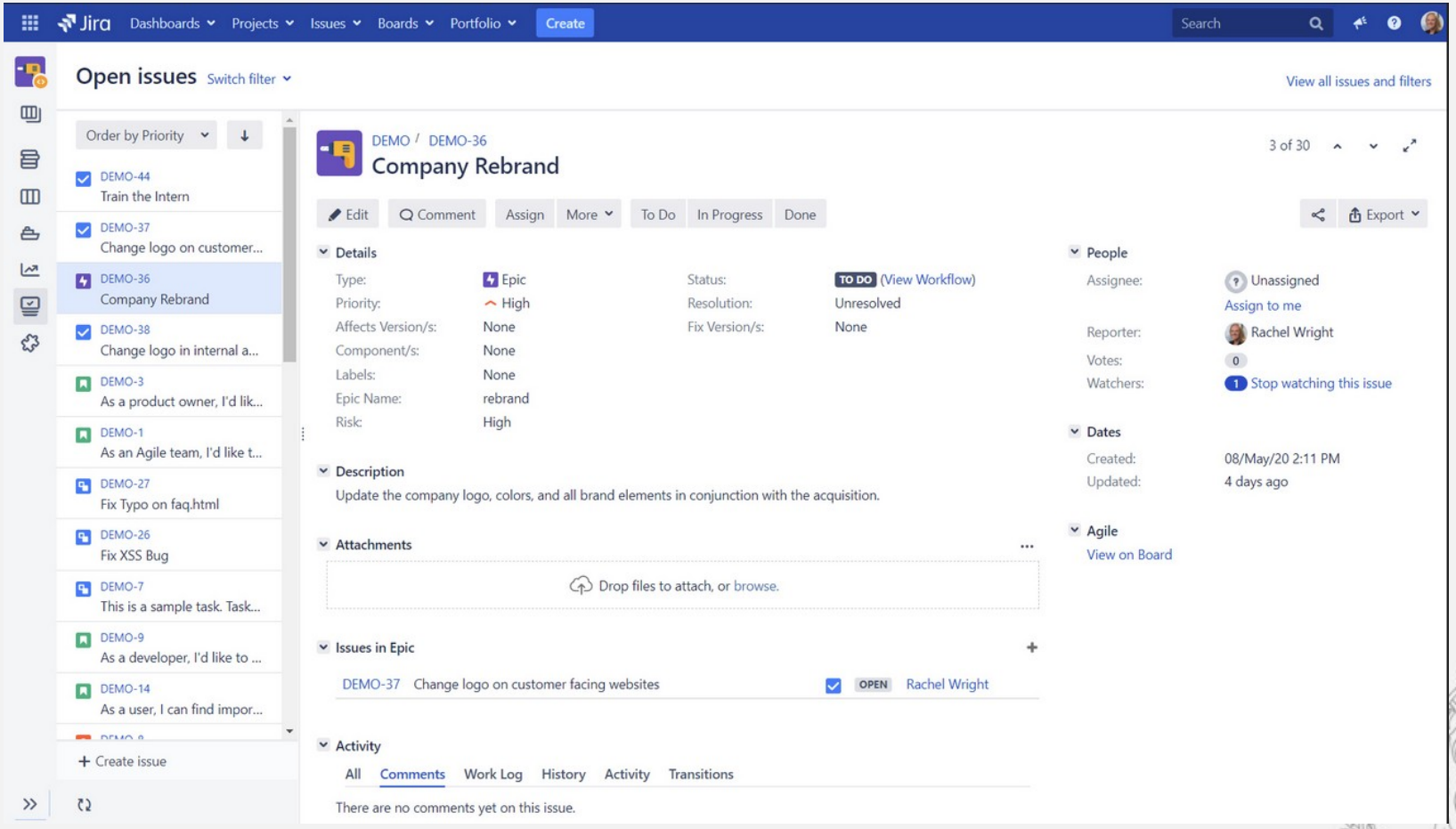
DDD tactic



Goal tracking



Issue tracking



Potremmo continuare

se solo potessimo fermare il tempo ....

InformAtelier

Riassunto

- Documentazione è importante.
- Documentare è importante.
- Scegliere il giusto mix è fondamentale.

Share Stories

- Uri Hasson:

  “Story telling fuse the present and the past to reshape our future brains.”

Quindi

- “Put the right letters together and make a better day”, Alphabet Street , Prince