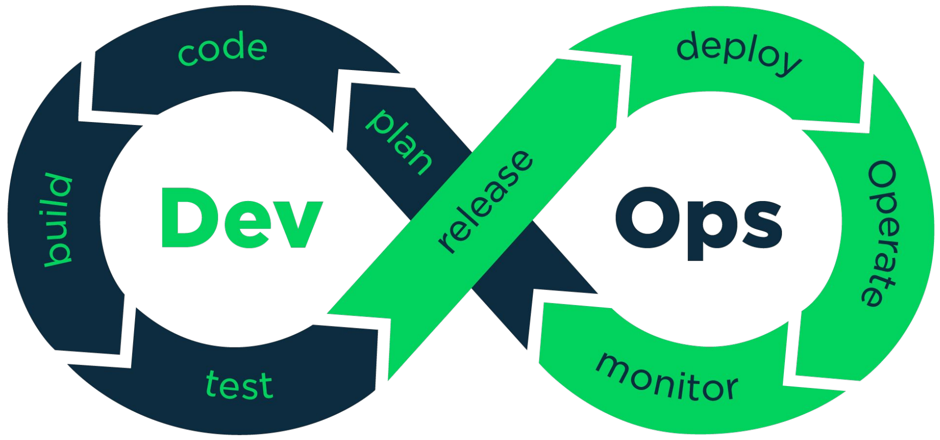
Documentation in Continuous Software Development - Not really required!?

Uwe van Heesch
Freelance Software Architect
Lecturer Software Engineering
19 December 2017

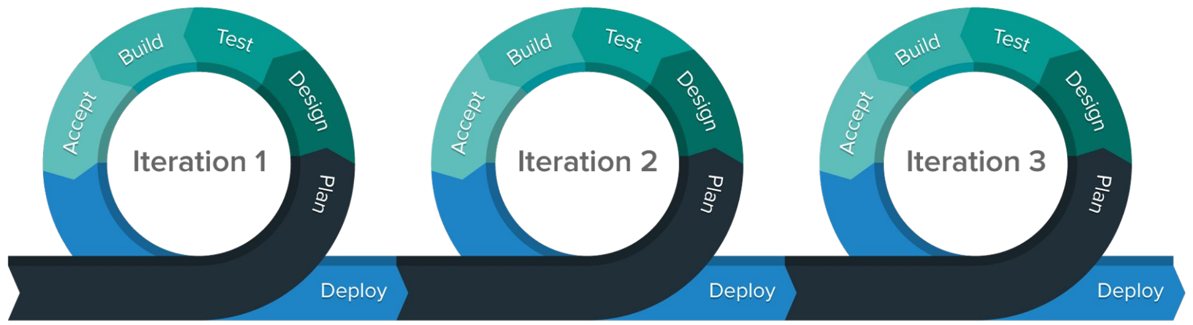
Hogeschool van Arnhem en Nijmegen
HAN University of Applied Sciences

*DIN EN ISO 7010, ASR A1.3 = RTFM!

What is Continuous Software Development?



Agile



Lean SD

ELIMINATE WASTE

AMPLIFY LEARNING

DECIDE AS LATE AS POSSIBLE

DELIVER AS FAST AS POSSIBLE

EMPOWER THE TEAM

BUILD QUALITY IN

SEE THE WHOLE

Principles of Continuous Software Development

1. **Strive for finding an optimal balance between effectiveness and efficiency**
   Eliminate Waste; measure Customer Satisfaction, continuous improvement using PIs

2. **Amplify learning and continuous improvement**
   Improve performance by reflection and feedback; share knowledge, tools and success

3. **Be flexible to adapt to new and unforeseen situations**
   Use flexible architectures, decide as late as possible, welcome changes, de facto standards vs de jure standards

4. **Strive for fast time-to-market**
   Use high degree of automation, deliver as fast as possible, release frequently

Principles of Continuous Software Development

5. **Establish a community of trust within and between the development team and other parties**
   Empower the team, trusted individuals, team culture

6. **Staff with competences**
   Team employs domain, technical and process-related skills, quality assurance

7. **Enable the team to see the big picture**
   Face-to-face conversation is richer than document-based conversation alone

8. **Collaborate and get involved**
   Do not work for the customer, work with the customer; establish shared principles, concerns, and priorities

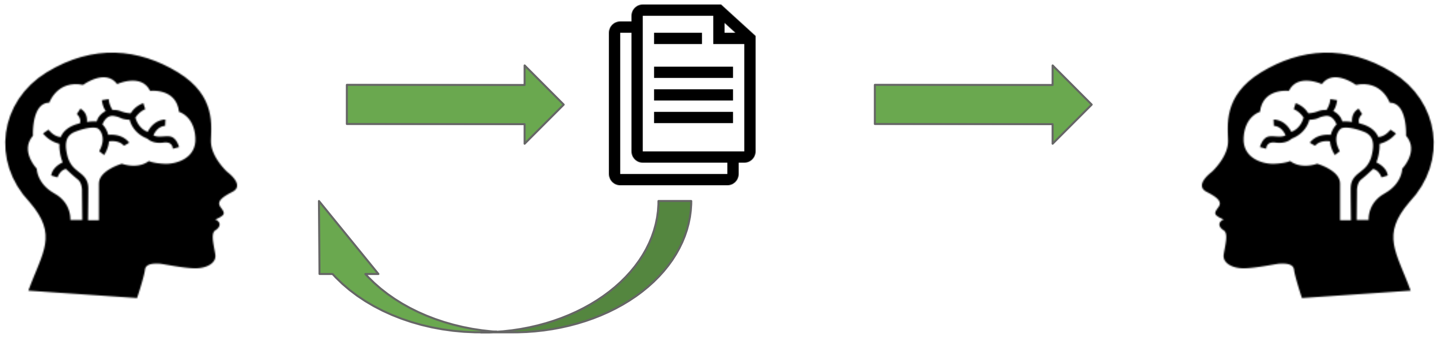
What is Documentation?

“A document is a written, drawn, or presented representation of thought. The word originates from the Latin documentum, which denotes a "teaching" or "lesson": the verb doceo denotes ‘to teach’.” (Wikipedia)

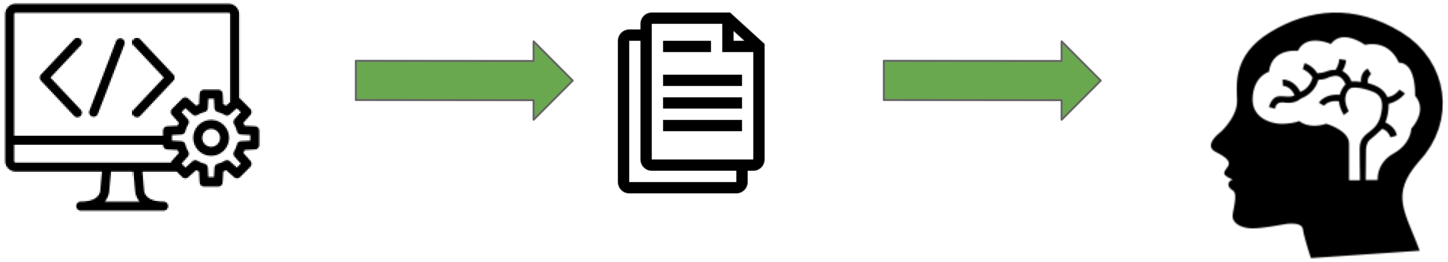
Documents are used to convey information to a specific audience for the purpose of teaching.

2 Categories of documentation in SW

1. Documents conveying a vision, plan or instructions



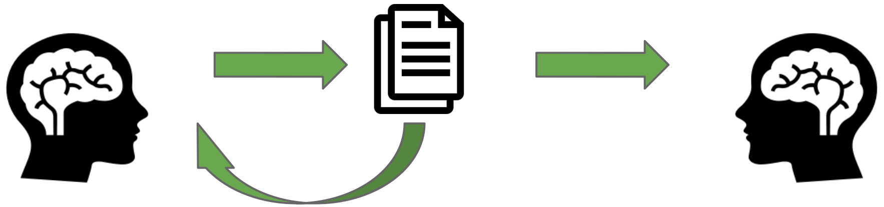
2. Documents explaining produced software-related artifacts



2 Categories of documentation in SW

Documents conveying a vision, plan or instructions, e.g.

- Requirements
  - User Stories, Use Cases, Business rules, Constraints
  - Quality Attribute Requirements (scalability, performance, security,...)
  - Regulations
- Up-front Specifications
  - Architectural decisions, architectural and detailed design, SW-design sketches on a whiteboard
  - UI sketches, Acceptance test-cases, Interface specifications
- Standards
  - Coding Standards, Quality Guidelines, Templates, Conventions,...
- ...



2 Categories of documentation in SW

Documents explaining produced software-related artifacts

- Deployment/ installation instructions
- Architectural decisions realized
- Architectural and detailed design realized
- SW-Interface specifications realized
- Documentation required for getting market admission
- Documentation as a contractual deliverable
- End-user manuals
- Training-material
- ...

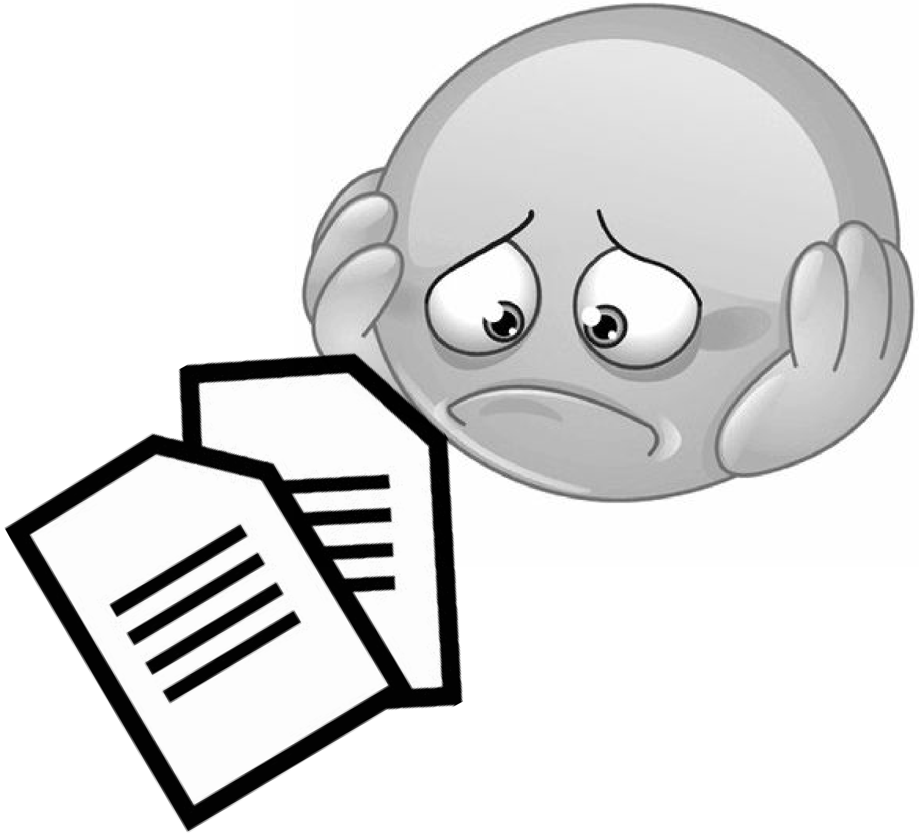
We have an odd relationship with documentation

TM;DW

NT;DW

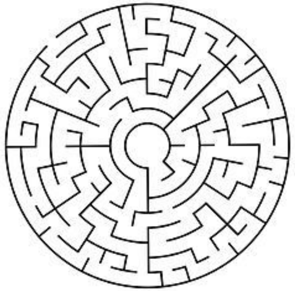
NS;DW

NM;DW



TL;DR

Documentation Smells



Version maze: Documentation is not explicit about the version of the software it describes. It is then unclear to the reader whether the information provided is still up-to-date. Even worse, documentation is sometimes compiled of items that describe different versions.

Cyclic references: Documentation items often consist of multiple items or documents referencing each other intensively. Sometimes, important aspects are not thoroughly described, because documentation items reference each other in a cyclic way without actually providing the information at some place.

Documentation Smells

Incompleteness: Documentation contains "TODOs" and gaps, i.e. parts of the system, or aspects, that are not described. When wikis are used, empty pages or page stubs with "under construction" as only content are a symptom of this issue. Also related: documentation scattered all over the place

Copy/paste: Documentation writers take over large large snippets of code or design artifacts that are too detailed to effectively serve as documentation. Sometimes, production data ends up in external documentation by accident; usage of text that violates Netiquette or is not politically correct has also been reported and qualifies as a specification/documentation smell as well.

Documentation Smells

Planet Power-Point: Documentation is prepared in form of a slide-based presentation. Stakeholders appreciate short summaries with a language they understand. But such presentations are often too abstract for many purposes (especially for developers and operators). The slides alone are ambiguous and require the “voice-over” to be understood.

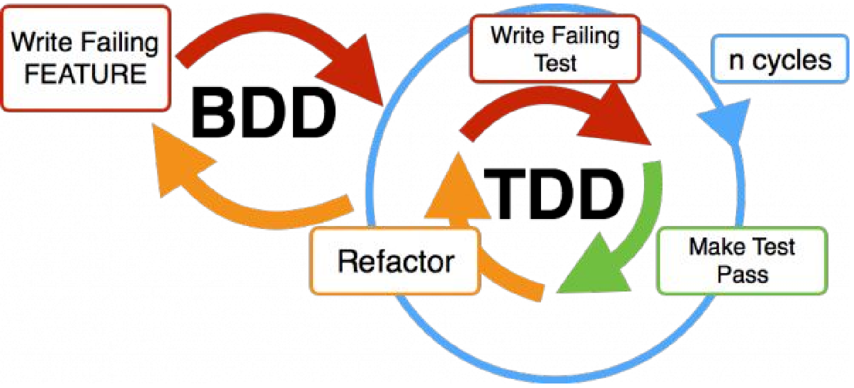
Zombie specifications/ aka dead documents: Documents do not have any readership and were not updated in a long time. You can tell from missing references to them in meetings and other specification/documentation items. Such items qualify as waste; they should either be updated and improved to meet an information need in a particular target audience, marked as ”stalled” and archived, or discarded.

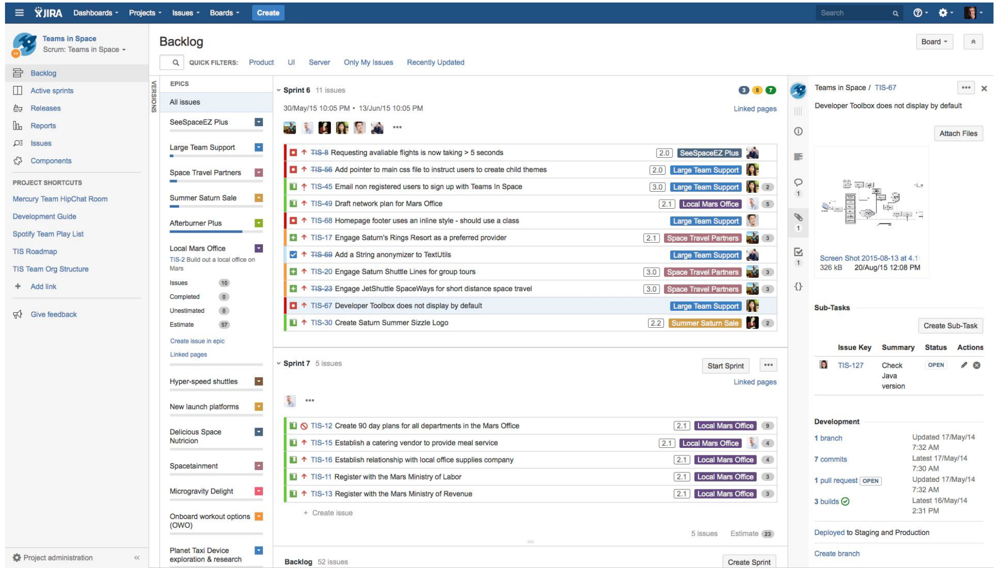
Violation of the ConSD principles

- Some effort is spent, but goals are not achieved => Lose/lose situation regarding efficiency and effectiveness
- Learning not amplified but hindered
- Documentation is often not flexible, but rigid
- Careless documentation causes loss of trust

So what’s the cure?

ConSD offers unique opportunities for making documentation efficient and effective





Teams in Space
Scrum: Teams in Space

Backlog
Active sprints
Releases
Reports
Issues
Components

PROJECT SHORTCUTS
Mercury Team HipChat Room
Development Guide
Spotify Team Play List
TIS Roadmap
TIS Team Org Structure
+ Add link

Give feedback

Project administration

Backlog

QUICK FILTERS: Product UI Server Only My Issues Recently Updated

VERSIONS

EPICS
All issues
SeeSpaceEZ Plus
Large Team Support
Space Travel Partners
Summer Saturn Sale
Afterburner Plus
Local Mars Office
TIS-2 Build out a local office on Mars

Issues	10
Completed	0
Unestimated	0
Estimate	57

Create issue in epic
Linked pages

Hyper-speed shuttles
New launch platforms
Delicious Space Nutricion
Spacetainment
Microgravity Delight
Onboard workout options (OWO)
Planet Taxi Device exploration & research

Sprint 6 11 issues 3 8 7
30/May/15 10:05 PM • 13/Jun/15 10:05 PM
Linked pages

~~TIS-8~~ Requesting avaliable flights is now taking > 5 seconds	2.0	SeeSpaceEZ Plus	
~~TIS-56~~ Add pointer to main css file to instruct users to create child themes	2.0	Large Team Support	
TIS-45 Email non registered users to sign up with Teams In Space	3.0	Large Team Support	2
TIS-49 Draft network plan for Mars Office	2.1	Local Mars Office	5
TIS-68 Homepage footer uses an inline style - should use a class		Large Team Support	
TIS-17 Engage Saturn's Rings Resort as a preferred provider	2.1	Space Travel Partners	3
~~TIS-69~~ Add a String anonymizer to TextUtils		Large Team Support	
TIS-20 Engage Saturn Shuttle Lines for group tours	3.0	Space Travel Partners	3
~~TIS-23~~ Engage JetShuttle SpaceWays for short distance space travel	3.0	Space Travel Partners	3
TIS-67 Developer Toolbox does not display by default		Large Team Support	
TIS-30 Create Saturn Summer Sizzle Logo	2.2	Summer Saturn Sale	2

Sprint 7 5 issues
Start Sprint
Linked pages

TIS-12 Create 90 day plans for all departments in the Mars Office	2.1	Local Mars Office	9
TIS-15 Establish a catering vendor to provide meal service	2.1	Local Mars Office	4
TIS-16 Establish relationship with local office supplies company	2.1	Local Mars Office	4
TIS-11 Register with the Mars Ministry of Labor	2.1	Local Mars Office	3
TIS-13 Register with the Mars Ministry of Revenue	2.1	Local Mars Office	3

+ Create issue

5 issues Estimate 23

Backlog 52 issues
Create Sprint

Teams in Space / TIS-67
Developer Toolbox does not display by default
Attach Files

Screen Shot 2015-08-13 at 4.1
326 kB 20/Aug/15 12:08 PM

Sub-Tasks
Create Sub-Task

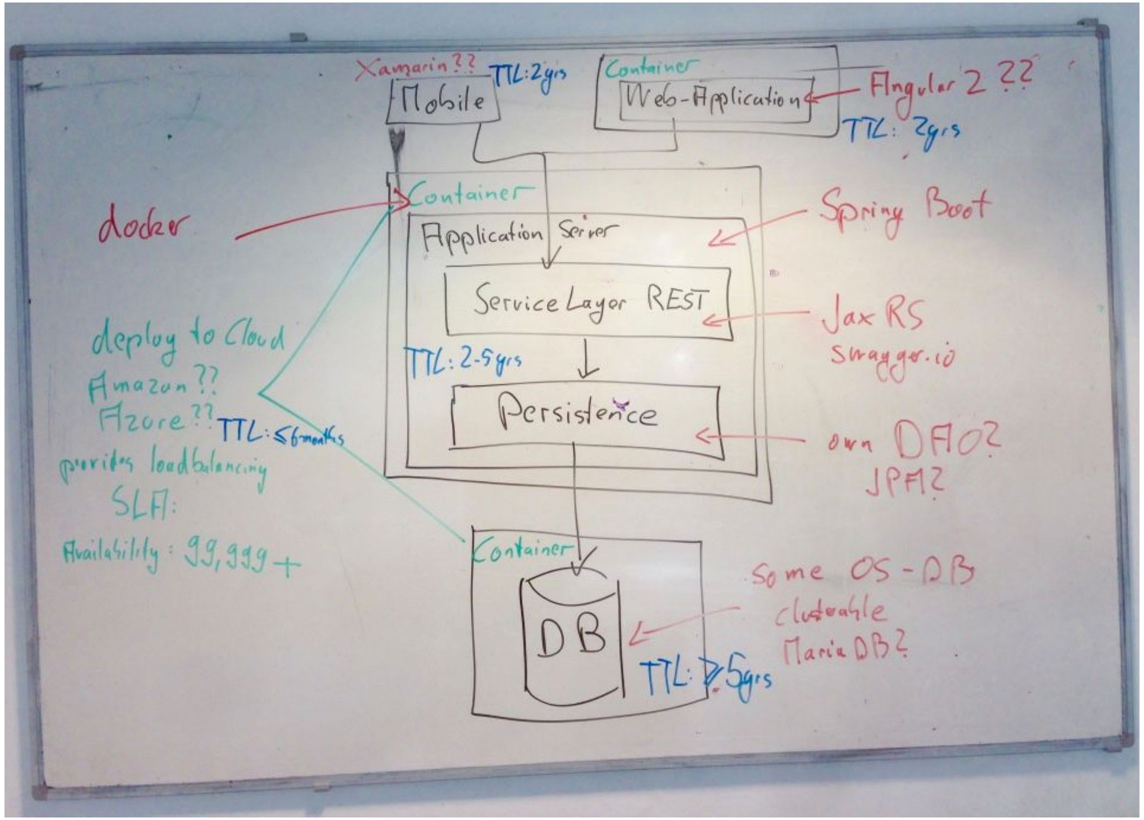
Issue Key	Summary	Status	Actions
TIS-127	Check Java version	OPEN	

Development

1 branch	Updated 17/May/14 7:32 AM
7 commits	Latest 17/May/14 7:30 AM
1 pull request OPEN	Updated 17/May/14 7:32 AM
3 builds	Latest 16/May/14 2:31 PM

Deployed to Staging and Production
Create branch

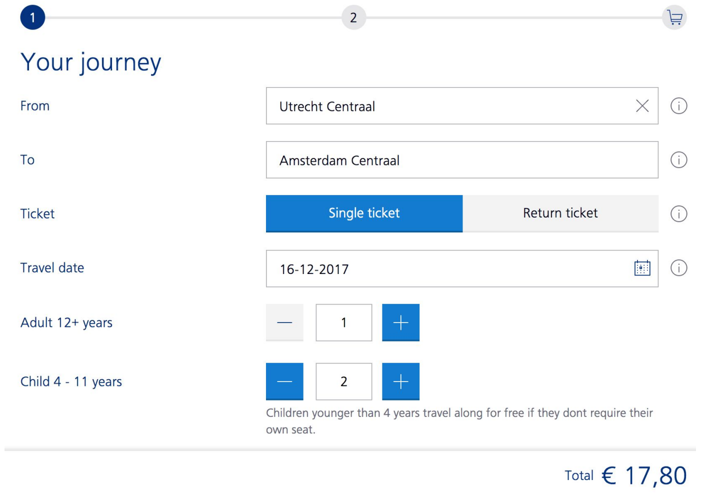
Use verbal communication and sketches for conveying visions and thoughts



TALKING HEADS

Use executable specifications for all the rest

Executable test-cases as functional requirements

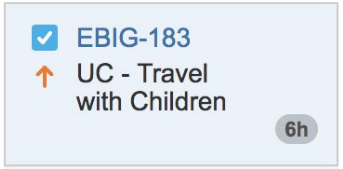


Your journey

From	Utrecht Centraal
To	Amsterdam Centraal
Ticket	Single ticket / Return ticket
Travel date	16-12-2017
Adult 12+ years	1
Child 4 - 11 years	2

Children younger than 4 years travel along for free if they dont require their own seat.

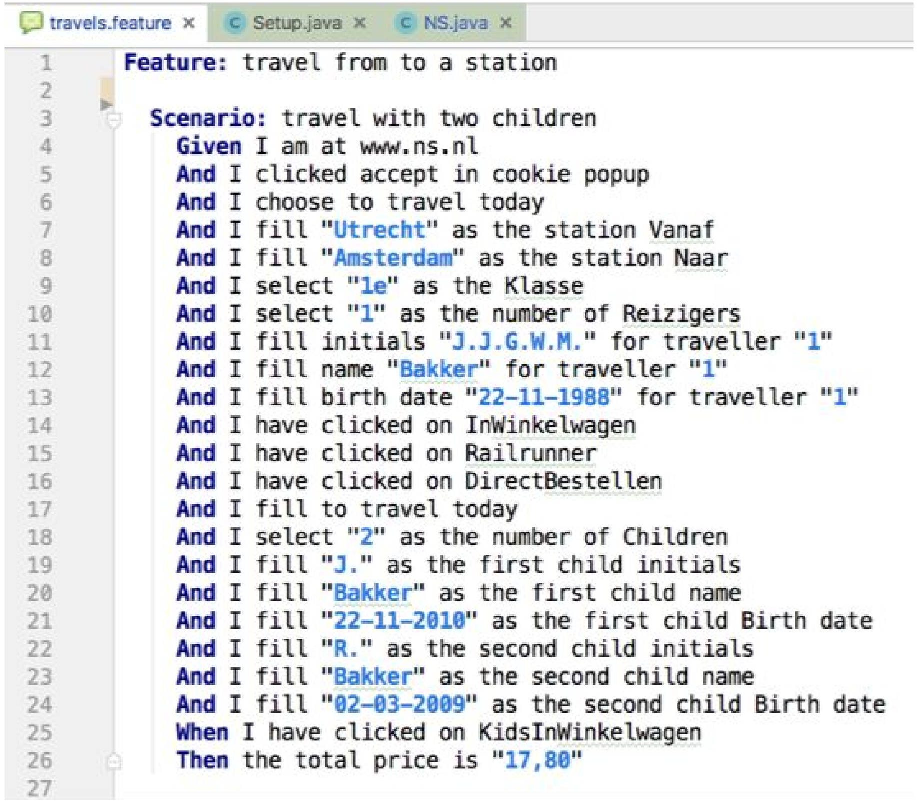
Total € 17,80



EBIG-183

UC - Travel with Children

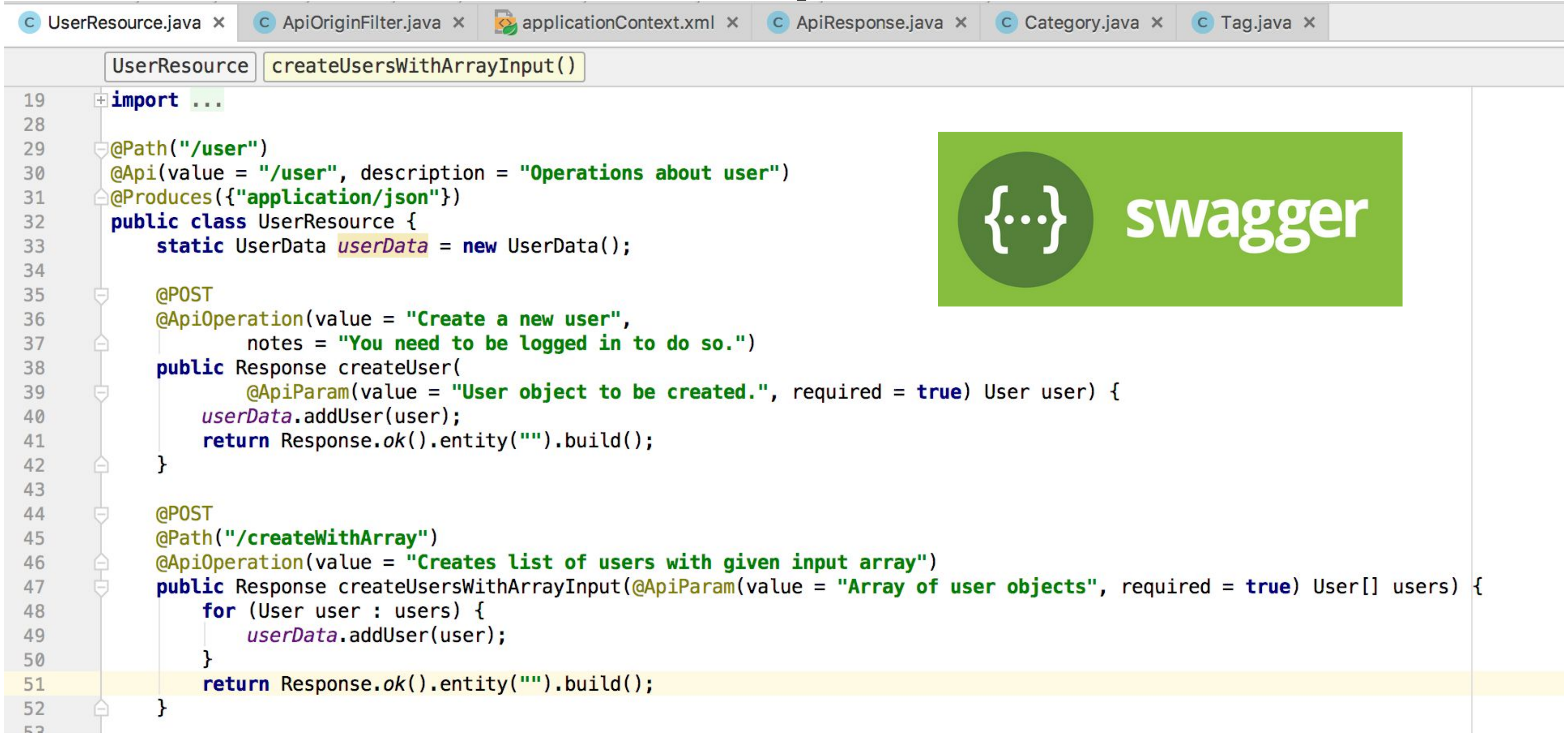
6h



```
Feature: travel from to a station

  Scenario: travel with two children
    Given I am at www.ns.nl
    And I clicked accept in cookie popup
    And I choose to travel today
    And I fill "Utrecht" as the station Vanaf
    And I fill "Amsterdam" as the station Naar
    And I select "1e" as the Klasse
    And I select "1" as the number of Reizigers
    And I fill initials "J.J.G.W.M." for traveller "1"
    And I fill name "Bakker" for traveller "1"
    And I fill birth date "22-11-1988" for traveller "1"
    And I have clicked on InWinkelwagen
    And I have clicked on Railrunner
    And I have clicked on DirectBestellen
    And I fill to travel today
    And I select "2" as the number of Children
    And I fill "J." as the first child initials
    And I fill "Bakker" as the first child name
    And I fill "22-11-2010" as the first child Birth date
    And I fill "R." as the second child initials
    And I fill "Bakker" as the second child name
    And I fill "02-03-2009" as the second child Birth date
    When I have clicked on KidsInWinkelwagen
    Then the total price is "17,80"

```

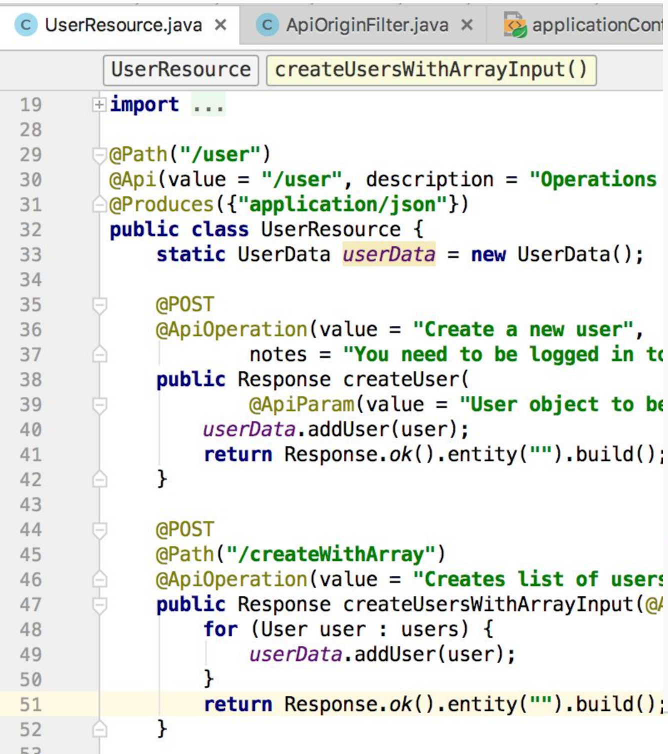
cucumber

Executable API Interface Specifications

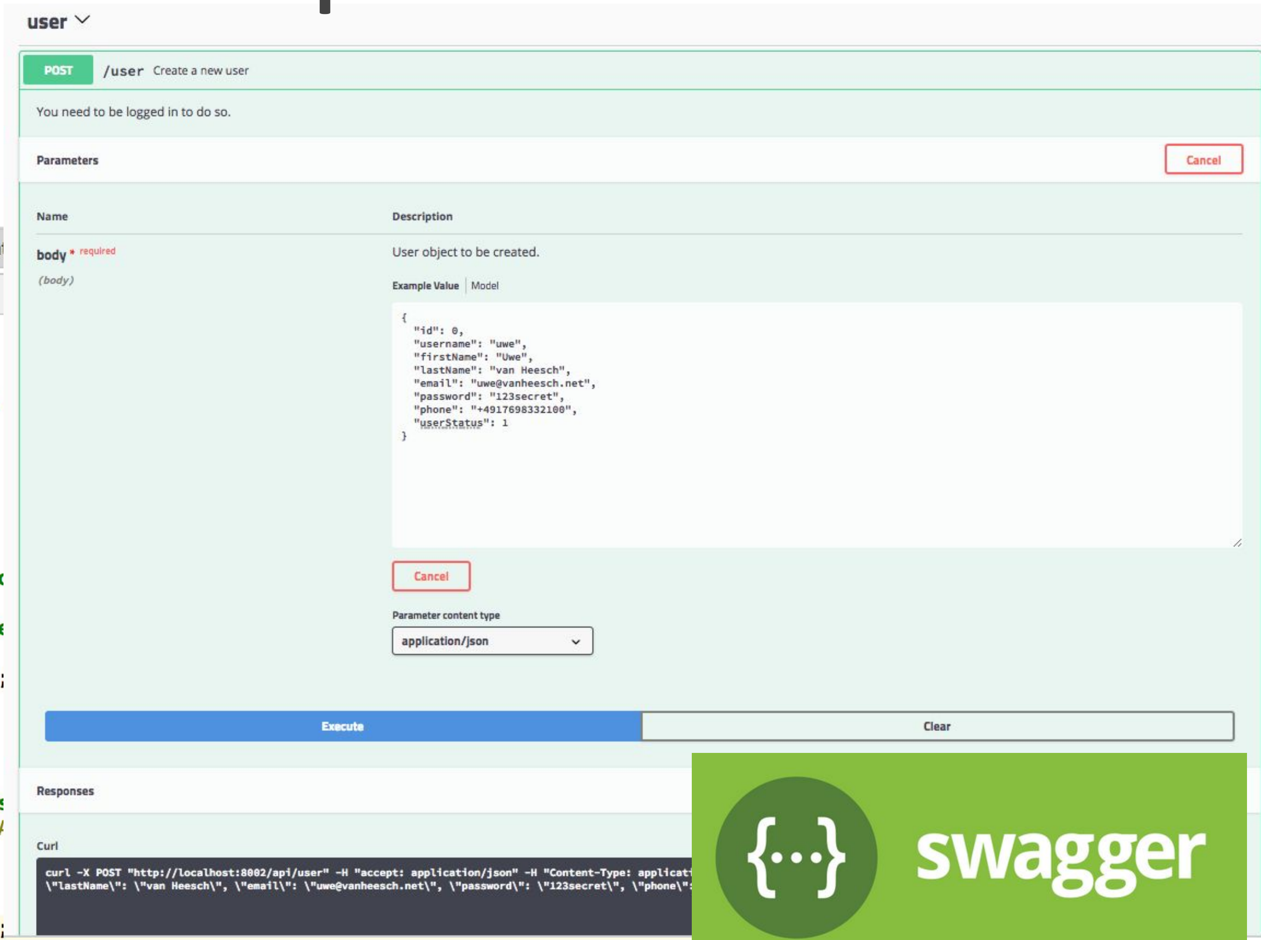


```java
import ...

@Path("/user")
@Api(value = "/user", description = "Operations about user")
@Produces({"application/json"})
public class UserResource {
    static UserData userData = new UserData();

    @POST
    @ApiOperation(value = "Create a new user",
            notes = "You need to be logged in to do so.")
    public Response createUser(
            @ApiParam(value = "User object to be created.", required = true) User user) {
        userData.addUser(user);
        return Response.ok().entity("").build();
    }

    @POST
    @Path("/createWithArray")
    @ApiOperation(value = "Creates list of users with given input array")
    public Response createUsersWithArrayInput(@ApiParam(value = "Array of user objects", required = true) User[] users) {
        for (User user : users) {
            userData.addUser(user);
        }
        return Response.ok().entity("").build();
    }
```

Executable API Interface Specifications



```java
import ...

@Path("/user")
@Api(value = "/user", description = "Operations
@Produces({"application/json"})
public class UserResource {
    static UserData userData = new UserData();

    @POST
    @ApiOperation(value = "Create a new user",
            notes = "You need to be logged in to
    public Response createUser(
            @ApiParam(value = "User object to be
        userData.addUser(user);
        return Response.ok().entity("").build();
    }

    @POST
    @Path("/createWithArray")
    @ApiOperation(value = "Creates list of users
    public Response createUsersWithArrayInput(@A
        for (User user : users) {
            userData.addUser(user);
        }
        return Response.ok().entity("").build();
    }
```



/user Create a new user

You need to be logged in to do so.

Parameters

Name	Description
body * required (body)	User object to be created.

```json
{
  "id": 0,
  "username": "uwe",
  "firstName": "Uwe",
  "lastName": "van Heesch",
  "email": "uwe@vanheesch.net",
  "password": "123secret",
  "phone": "+4917698332100",
  "userStatus": 1
}
```

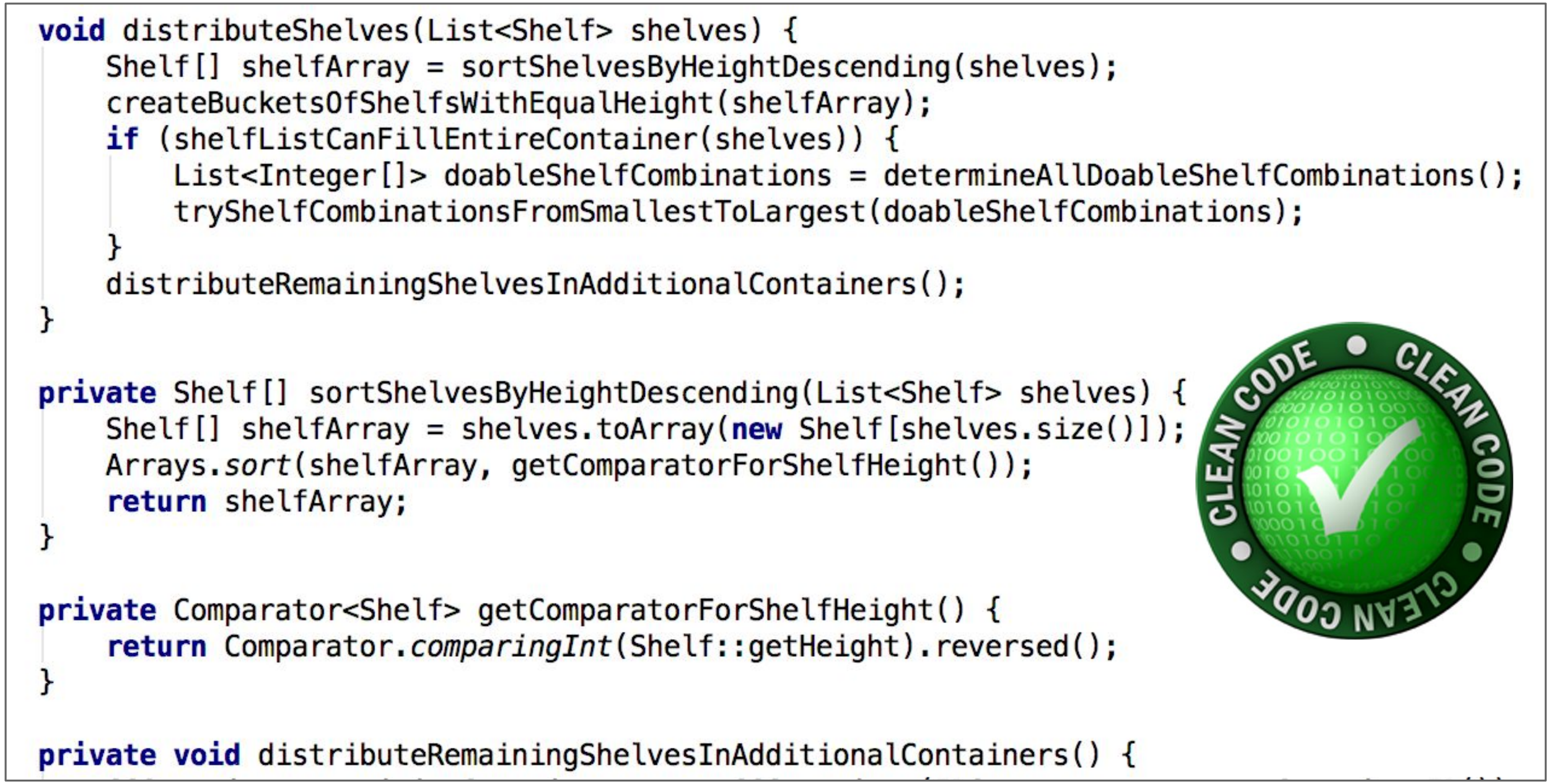
Parameter content type: application/json

Responses

Curl

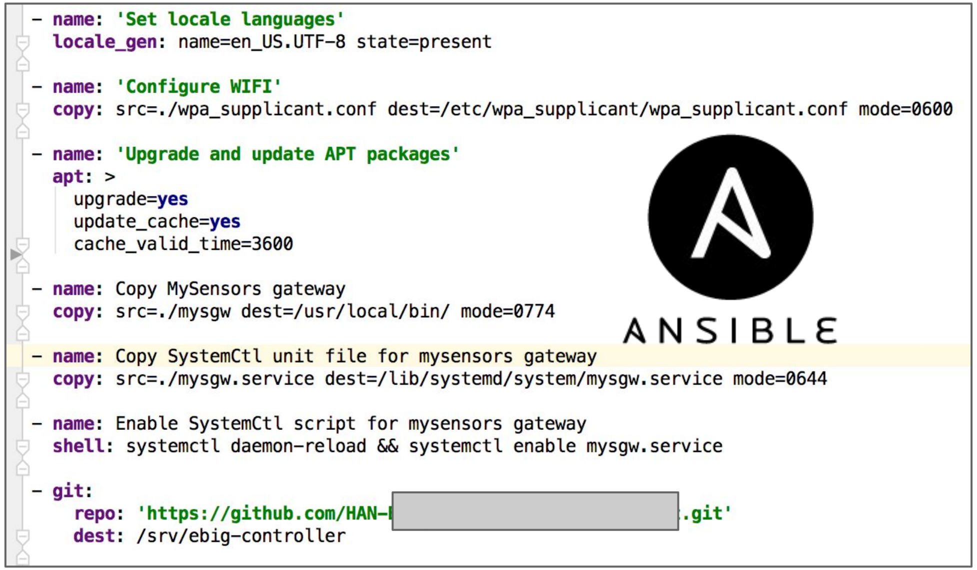
```
curl -X POST "http://localhost:8002/api/user" -H "accept: application/json" -H "Content-Type: applicati
\"lastName\": \"van Heesch\", \"email\": \"uwe@vanheesch.net\", \"password\": \"123secret\", \"phone\":
```

Source-code (production or test) as documentation for algorithms

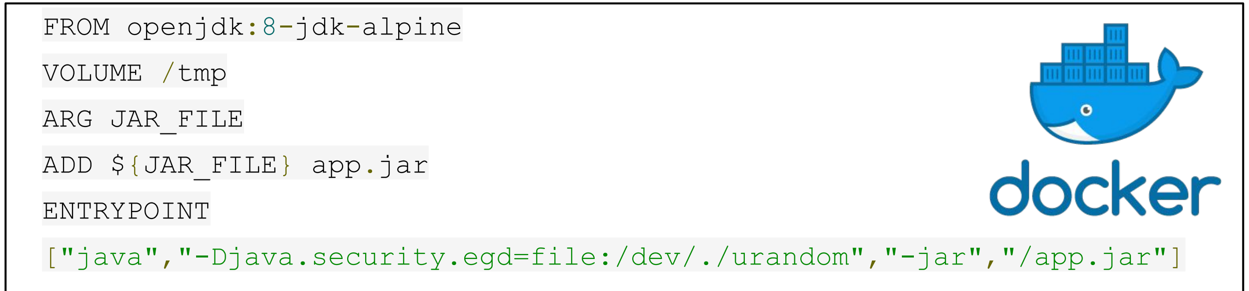


```java
void distributeShelves(List<Shelf> shelves) {
    Shelf[] shelfArray = sortShelvesByHeightDescending(shelves);
    createBucketsOfShelfsWithEqualHeight(shelfArray);
    if (shelfListCanFillEntireContainer(shelves)) {
        List<Integer[]> doableShelfCombinations = determineAllDoableShelfCombinations();
        tryShelfCombinationsFromSmallestToLargest(doableShelfCombinations);
    }
    distributeRemainingShelvesInAdditionalContainers();
}

private Shelf[] sortShelvesByHeightDescending(List<Shelf> shelves) {
    Shelf[] shelfArray = shelves.toArray(new Shelf[shelves.size()]);
    Arrays.sort(shelfArray, getComparatorForShelfHeight());
    return shelfArray;
}

private Comparator<Shelf> getComparatorForShelfHeight() {
    return Comparator.comparingInt(Shelf::getHeight).reversed();
}

private void distributeRemainingShelvesInAdditionalContainers() {
```

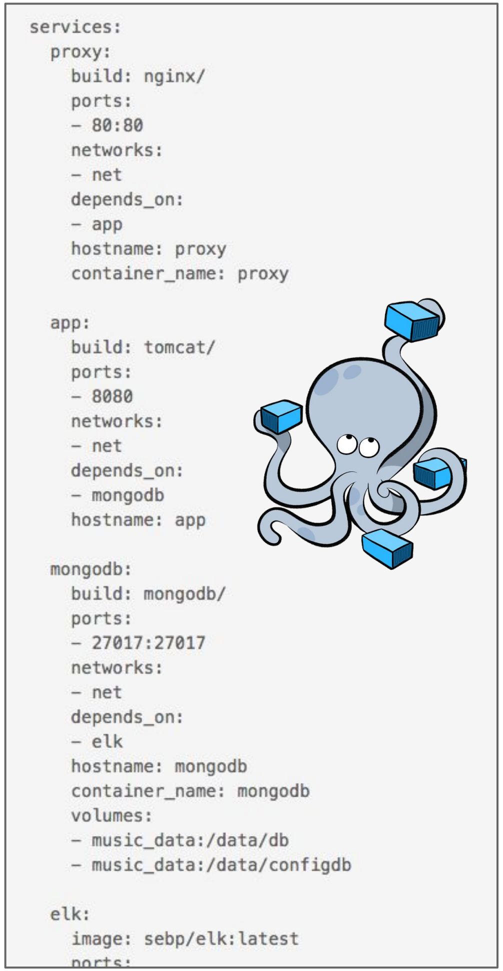
Build Server configuration and scripts as deployment documentation



```
- name: 'Set locale languages'
  locale_gen: name=en_US.UTF-8 state=present

- name: 'Configure WIFI'
  copy: src=./wpa_supplicant.conf dest=/etc/wpa_supplicant/wpa_supplicant.conf mode=0600

- name: 'Upgrade and update APT packages'
  apt: >
    upgrade=yes
    update_cache=yes
    cache_valid_time=3600

- name: Copy MySensors gateway
  copy: src=./mysgw dest=/usr/local/bin/ mode=0774

- name: Copy SystemCtl unit file for mysensors gateway
  copy: src=./mysgw.service dest=/lib/systemd/system/mysgw.service mode=0644

- name: Enable SystemCtl script for mysensors gateway
  shell: systemctl daemon-reload && systemctl enable mysgw.service

- git:
    repo: 'https://github.com/HAN-[redacted].git'
    dest: /srv/ebig-controller
```

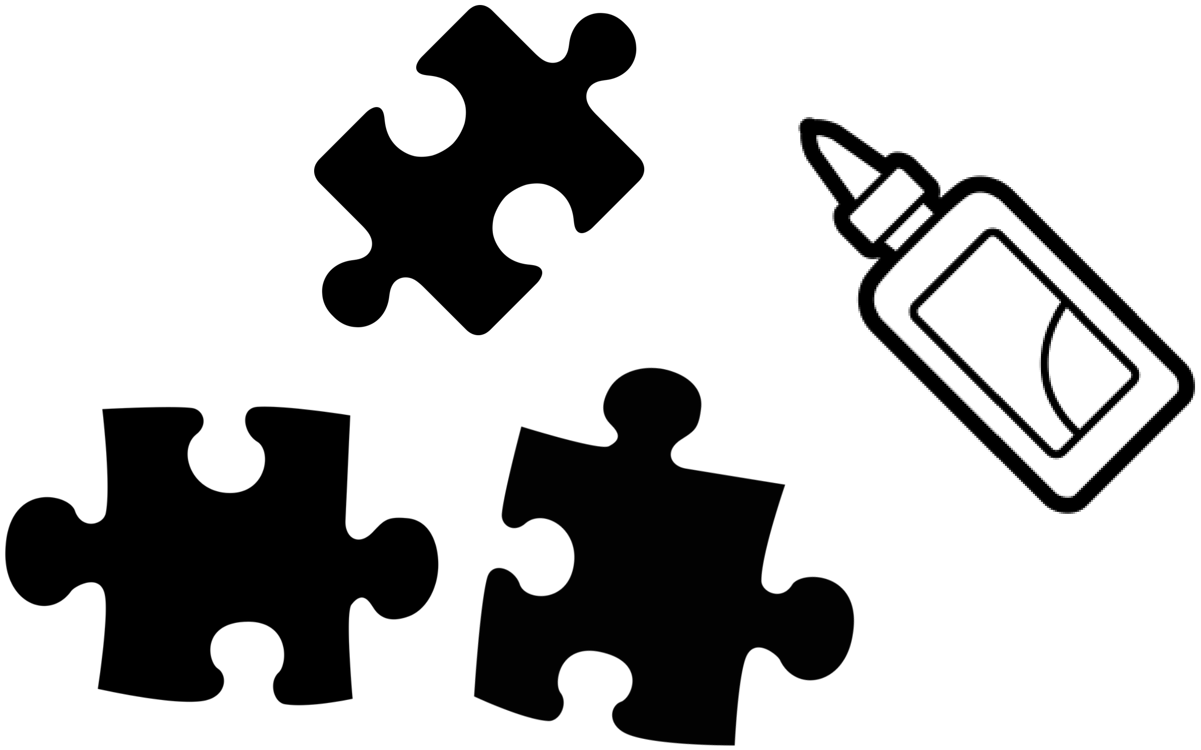


```
FROM openjdk:8-jdk-alpine
VOLUME /tmp
ARG JAR_FILE
ADD ${JAR_FILE} app.jar
ENTRYPOINT
["java","-Djava.security.egd=file:/dev/./urandom","-jar","/app.jar"]
```

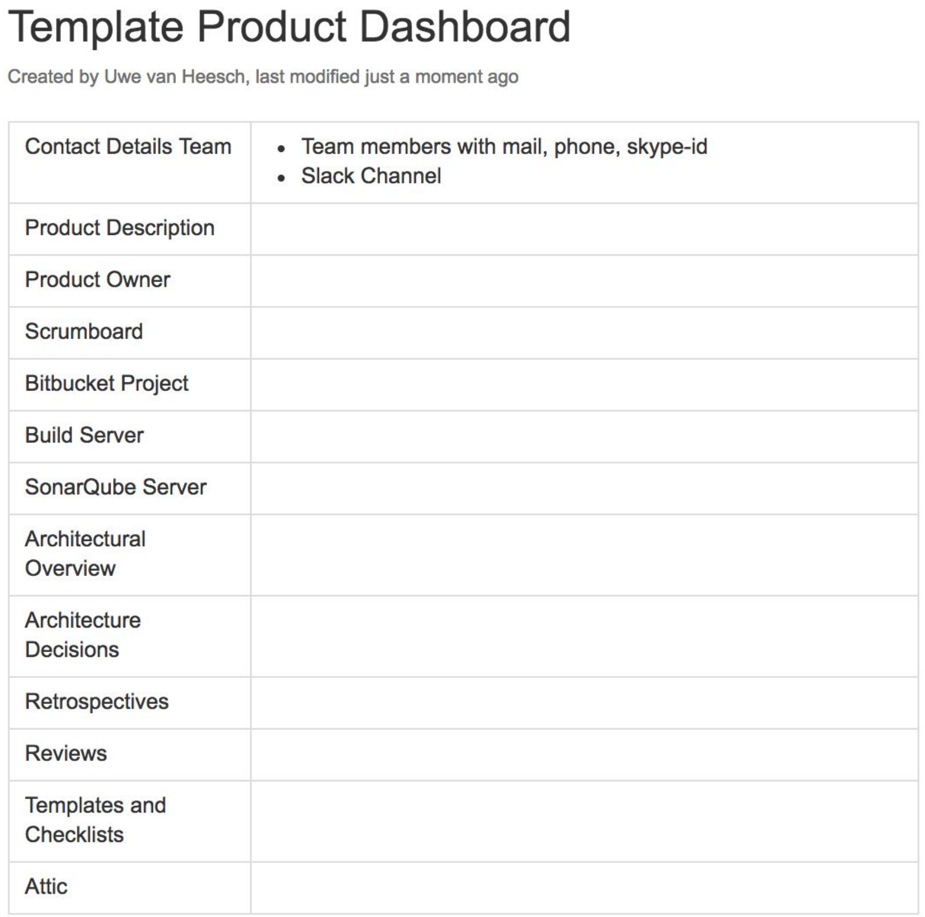


```
services:
  proxy:
    build: nginx/
    ports:
    - 80:80
    networks:
    - net
    depends_on:
    - app
    hostname: proxy
    container_name: proxy

  app:
    build: tomcat/
    ports:
    - 8080
    networks:
    - net
    depends_on:
    - mongodb
    hostname: app

  mongodb:
    build: mongodb/
    ports:
    - 27017:27017
    networks:
    - net
    depends_on:
    - elk
    hostname: mongodb
    container_name: mongodb
    volumes:
    - music_data:/data/db
    - music_data:/data/configdb

  elk:
    image: sebp/elk:latest
    ports:
```

How to glue everything together?



Single point of reference/ aka Yellow Pages

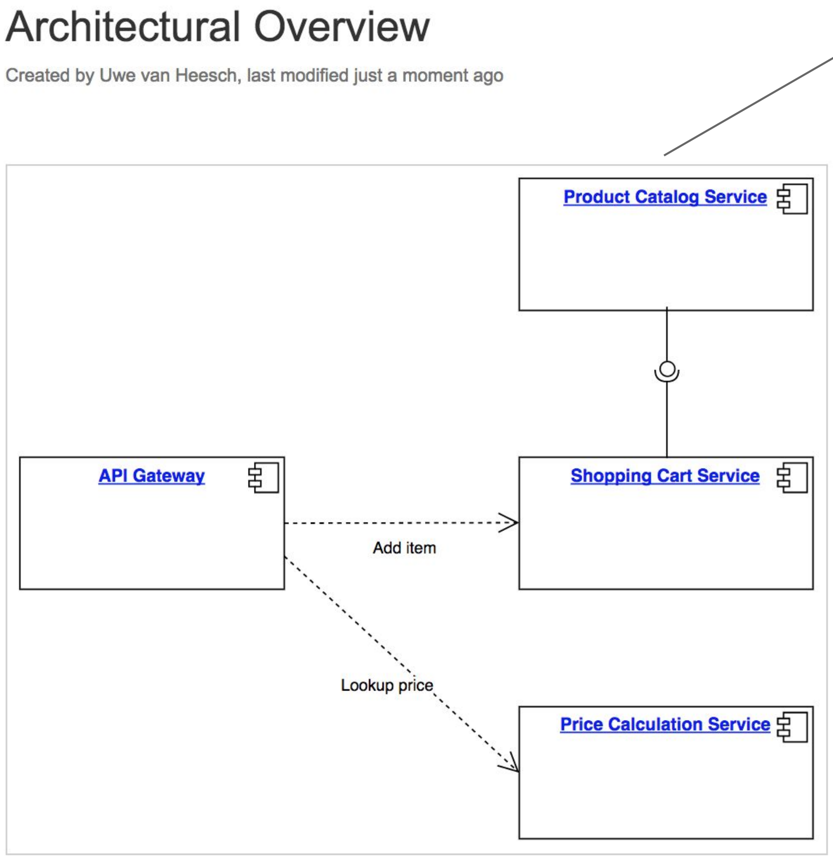


Template Product Dashboard

Created by Uwe van Heesch, last modified just a moment ago

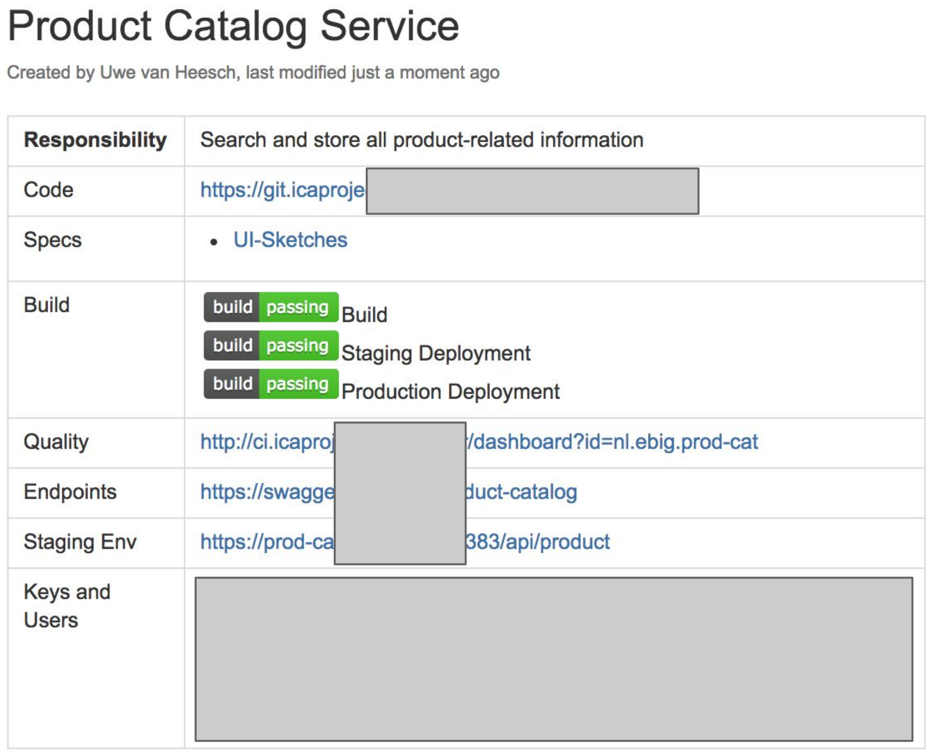
Contact Details Team	• Team members with mail, phone, skype-id • Slack Channel
Product Description	
Product Owner	
Scrumboard	
Bitbucket Project	
Build Server	
SonarQube Server	
Architectural Overview	
Architecture Decisions	
Retrospectives	
Reviews	
Templates and Checklists	
Attic	

Architecture Overview in a wiki



Architectural Overview

Created by Uwe van Heesch, last modified just a moment ago



Product Catalog Service

Created by Uwe van Heesch, last modified just a moment ago

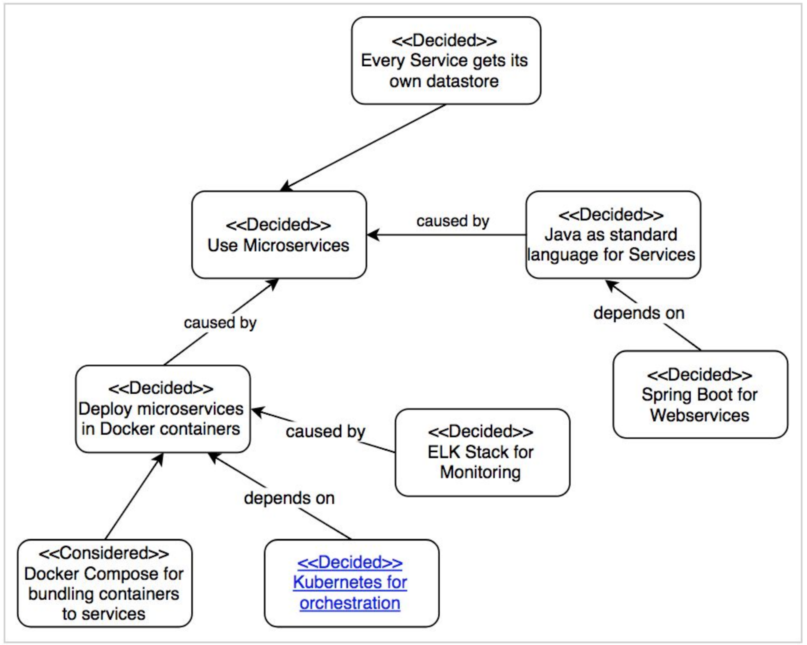
Responsibility	Search and store all product-related information
Code	https://git.icaproje [redacted]
Specs	• UI-Sketches
Build	build passing Build build passing Staging Deployment build passing Production Deployment
Quality	http://ci.icaproj [redacted] /dashboard?id=nl.ebig.prod-cat
Endpoints	https://swagge [redacted] duct-catalog
Staging Env	https://prod-ca [redacted] 383/api/product
Keys and Users	[redacted]

Document architecture decisions thoughtfully

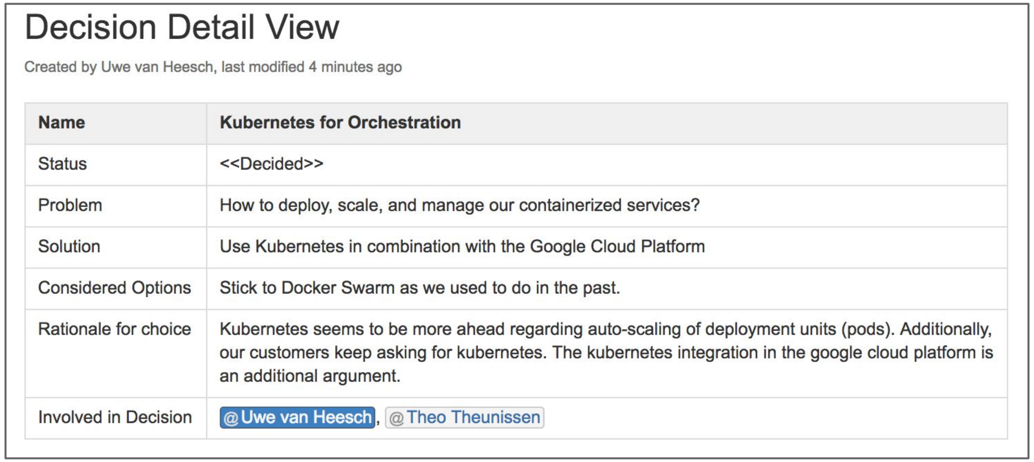
Pages / Product Dashboard   Edit   Save for later   Watching   Share   •••

Decision Relationship View

Created by Uwe van Heesch, last modified less than a minute ago



<<Decided>> Every Service gets its own datastore
<<Decided>> Use Microservices
caused by
<<Decided>> Java as standard language for Services
depends on
caused by
<<Decided>> Spring Boot for Webservices
<<Decided>> Deploy microservices in Docker containers
caused by
<<Decided>> ELK Stack for Monitoring
depends on
<<Considered>> Docker Compose for bundling containers to services
<<Decided>> Kubernetes for orchestration

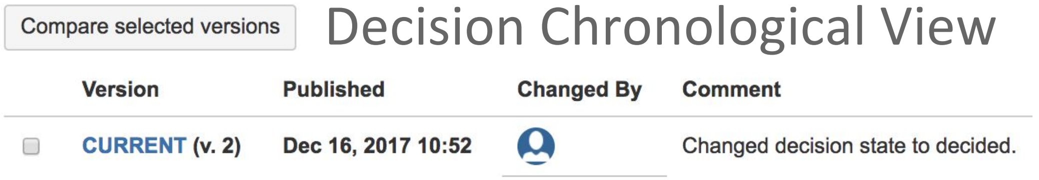


Decision Detail View

Created by Uwe van Heesch, last modified 4 minutes ago

Name	Kubernetes for Orchestration
Status	<<Decided>>
Problem	How to deploy, scale, and manage our containerized services?
Solution	Use Kubernetes in combination with the Google Cloud Platform
Considered Options	Stick to Docker Swarm as we used to do in the past.
Rationale for choice	Kubernetes seems to be more ahead regarding auto-scaling of deployment units (pods). Additionally, our customers keep asking for kubernetes. The kubernetes integration in the google cloud platform is an additional argument.
Involved in Decision	@Uwe van Heesch, @Theo Theunissen

Decision Chronological View



Compare selected versions

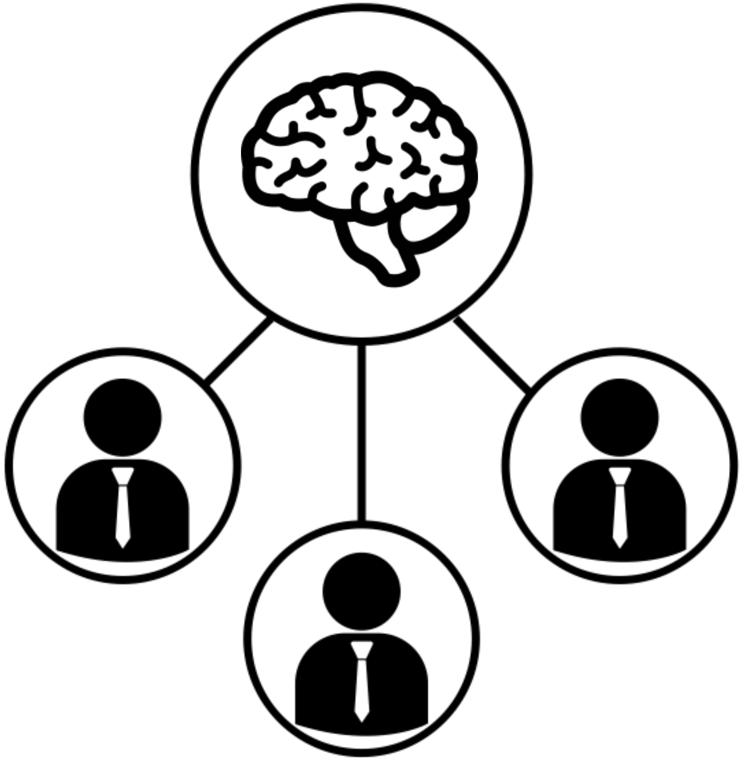
	Version	Published	Changed By	Comment
☐	CURRENT (v. 2)	Dec 16, 2017 10:52		Changed decision state to decided.

Like   Be the first to like this

Confluence

architecture   decisions

Exploit the fact that teams have a shared history and a common understanding



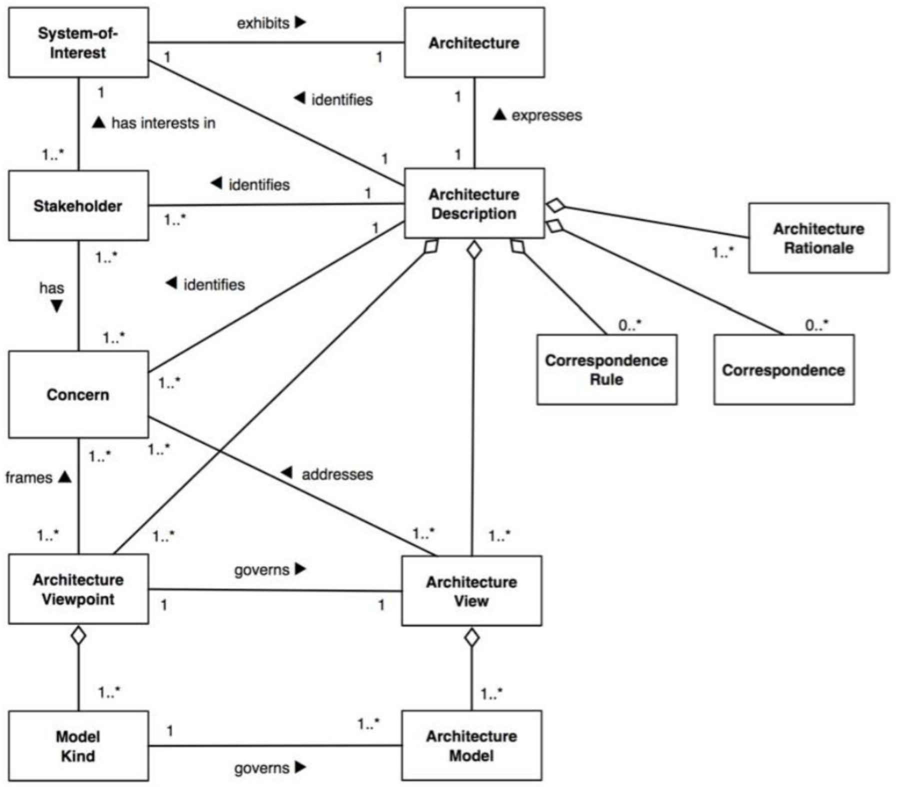
- Documentation by exception
- Spend effort where you derive from (de-facto) standards or common solutions
- Replace upfront documentation by realization
- Create documentation as a reading guide for source code/configuration
- DRY/ Open-Close/ Single Responsibility
- Refactor documentation, dare to throw documentation away

In reality we come across architecture documentation that is not (yet?) supported by tools to achieve executable specifications

- Business models
- Logical context of systems
- Non-technical views on architecture for business stakeholders
- Risk themes
- Decision rationale (could be achieved partially)
- Some quality attributes (e.g. learnability, operability, modifiability,..)
- Some domain-specific models (where DSLs are not appropriate)
- Also an issue: Finding the right level of abstraction for a specific target audience with a specific state of knowledge

Current research

Develop an architecture framework that supports the continuous transition from documentation to convey thoughts to documentation describing existing system artifacts



Theunissen, van Heesch, Avgeriou

ISO/IEC/IEEE 42010

>Manual< documentation in Continuous Software Development - Not really required >a lot<!

Uwe van Heesch
uwe@vanheesch.net
https://vanheesch.net

Further Readings

- van Heesch, U., et al. (2017). **Software specification in continuous software development** - A Focus Group Report. In Proceedings of the 22nd European Conference on Pattern Languages of Programs. ACM.

- Theunissen, T., & van Heesch, U. (2017). **Specification in Continuous Software Development**. In Proceedings of the 22nd European Conference on Pattern Languages of Programs. ACM.

- van Heesch, U., Avgeriou, P., & Hilliard, R. (2012). **A documentation framework for architecture decisions**. Journal of Systems and Software, 85(4), 795-820.

- van Heesch, U., Avgeriou, P., & Hilliard, R. (2012). **Forces on Architecture Decisions-A Viewpoint**. In Proceedings of the 2012 Joint Working Conference on Software Architecture and European Conference on Software Architecture. IEEE Computer Society.