A Process to Increase the Model Quality in the Context of Model-Based Testing

Vladimir Entin, Mathias Winder, Bo Zhang, Andreas Claus

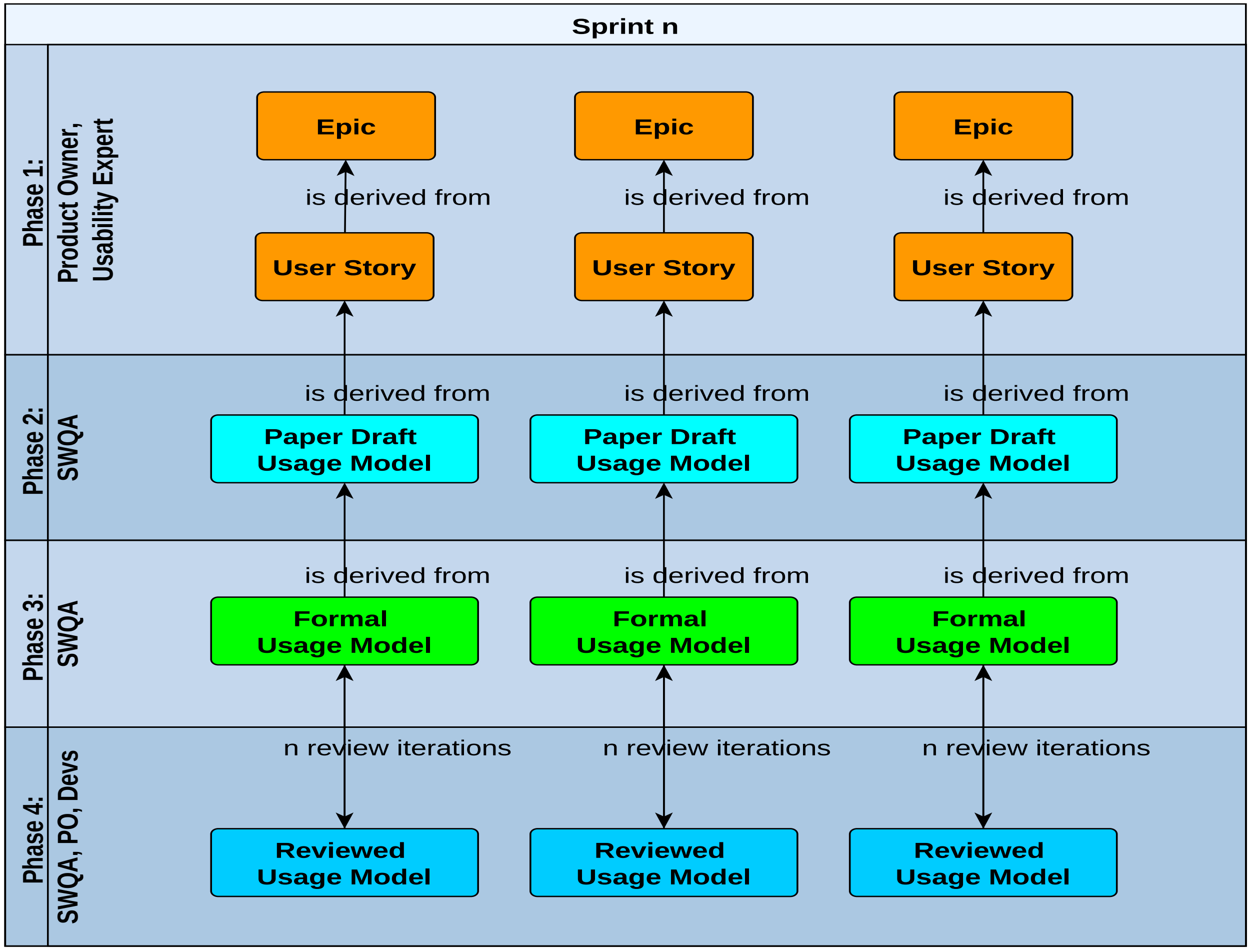
Graz, 17.04.2015

15 April 2015

Introduction

- OMICRON (www.omicron.at)
  - International company serving the electrical power industry with innovative testing and diagnostic solutions
  - Diagnostic of protection relays, instrument and power transformers
  - Located in Vorarlberg, Austria
  - ~ 700 employees
- Software is being written for our measurement devices
  - *Scaled Agile Framework* as development approach
  - 4 SCRUM teams -> 30 team members
  - .NET framework and C# for PC software
  - Mainly C++ for embedded software

Motivation

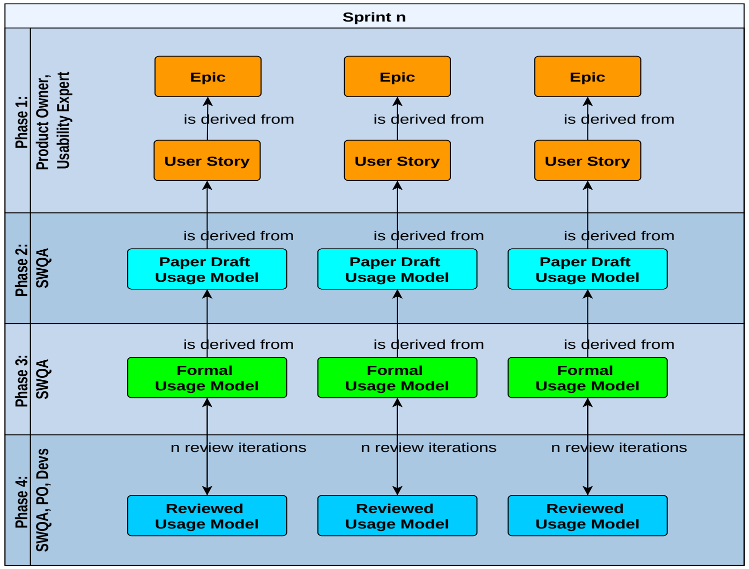

Sprint n
Phase 1: Product Owner, Usability Expert
Epic
is derived from
User Story
Epic
is derived from
User Story
Epic
is derived from
User Story
Phase 2: SWQA
is derived from
Paper Draft Usage Model
is derived from
Paper Draft Usage Model
is derived from
Paper Draft Usage Model
Phase 3: SWQA
is derived from
Formal Usage Model
is derived from
Formal Usage Model
is derived from
Formal Usage Model
Phase 4: SWQA, PO, Devs
n review iterations
Reviewed Usage Model
n review iterations
Reviewed Usage Model
n review iterations
Reviewed Usage Model

Challenges

- **How to** avoid information loss during the derivation of models from the textual requirements by the software quality engineer (SWQA)?

- **How to** reduce the maintainability effort of usage models?

- **How to** allow the less-experienced SWQA the application of model-based testing approaches?



Sprint n

Phase 1: Product Owner, Usability Expert

Epic | Epic | Epic

is derived from | is derived from | is derived from

User Story | User Story | User Story

Phase 2: SWQA

is derived from | is derived from | is derived from

Paper Draft Usage Model | Paper Draft Usage Model | Paper Draft Usage Model

Phase 3: SWQA

is derived from | is derived from | is derived from

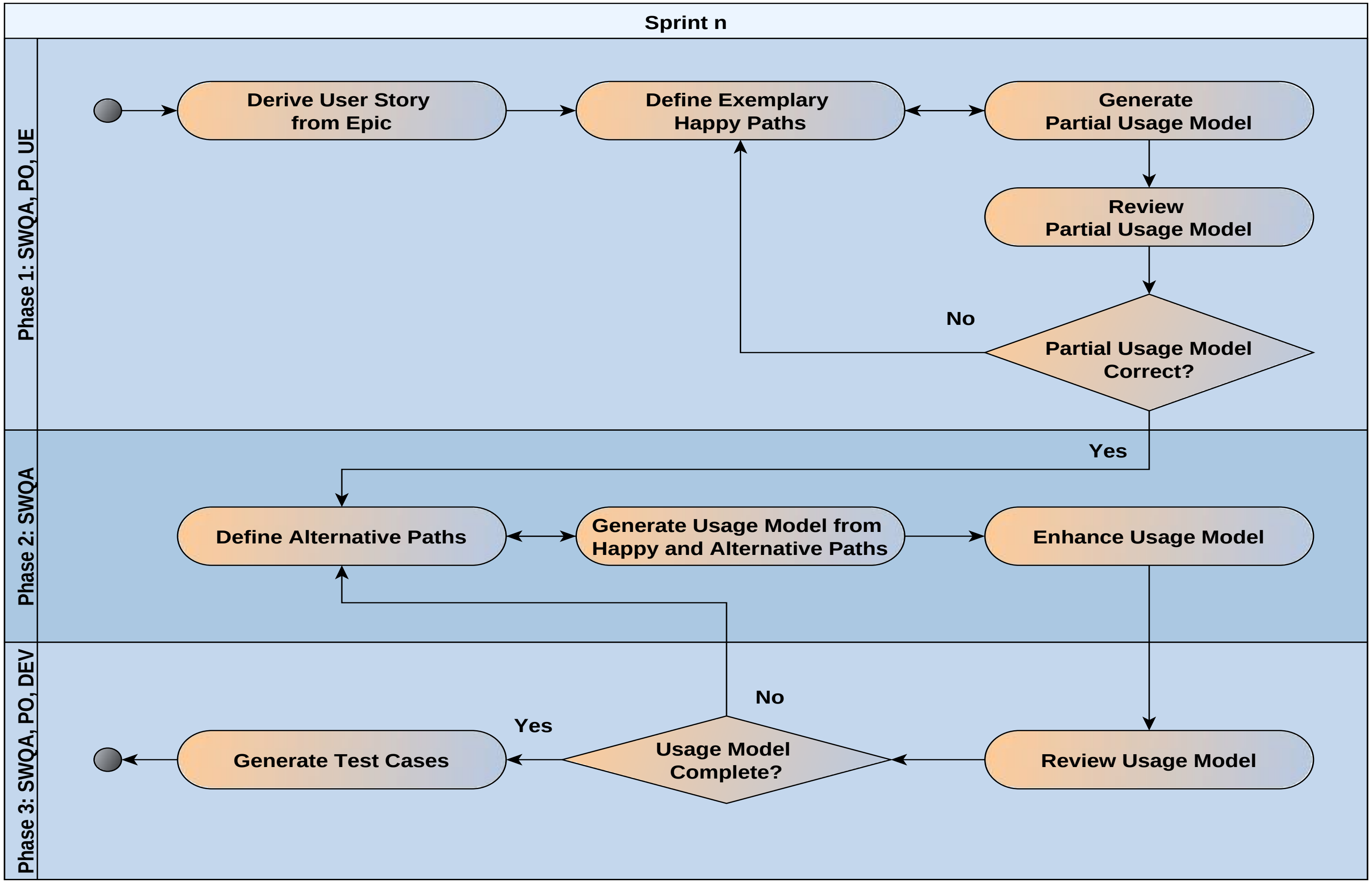
Formal Usage Model | Formal Usage Model | Formal Usage Model

Phase 4: SWQA, PO, Devs

n review iterations | n review iterations | n review iterations

Reviewed Usage Model | Reviewed Usage Model | Reviewed Usage Model

The Process


Sprint n
Phase 1: SWQA, PO, UE
Derive User Story from Epic
Define Exemplary Happy Paths
Generate Partial Usage Model
Review Partial Usage Model
No
Partial Usage Model Correct?
Yes
Phase 2: SWQA
Define Alternative Paths
Generate Usage Model from Happy and Alternative Paths
Enhance Usage Model
Phase 3: SWQA, PO, DEV
No
Yes
Generate Test Cases
Usage Model Complete?
Review Usage Model

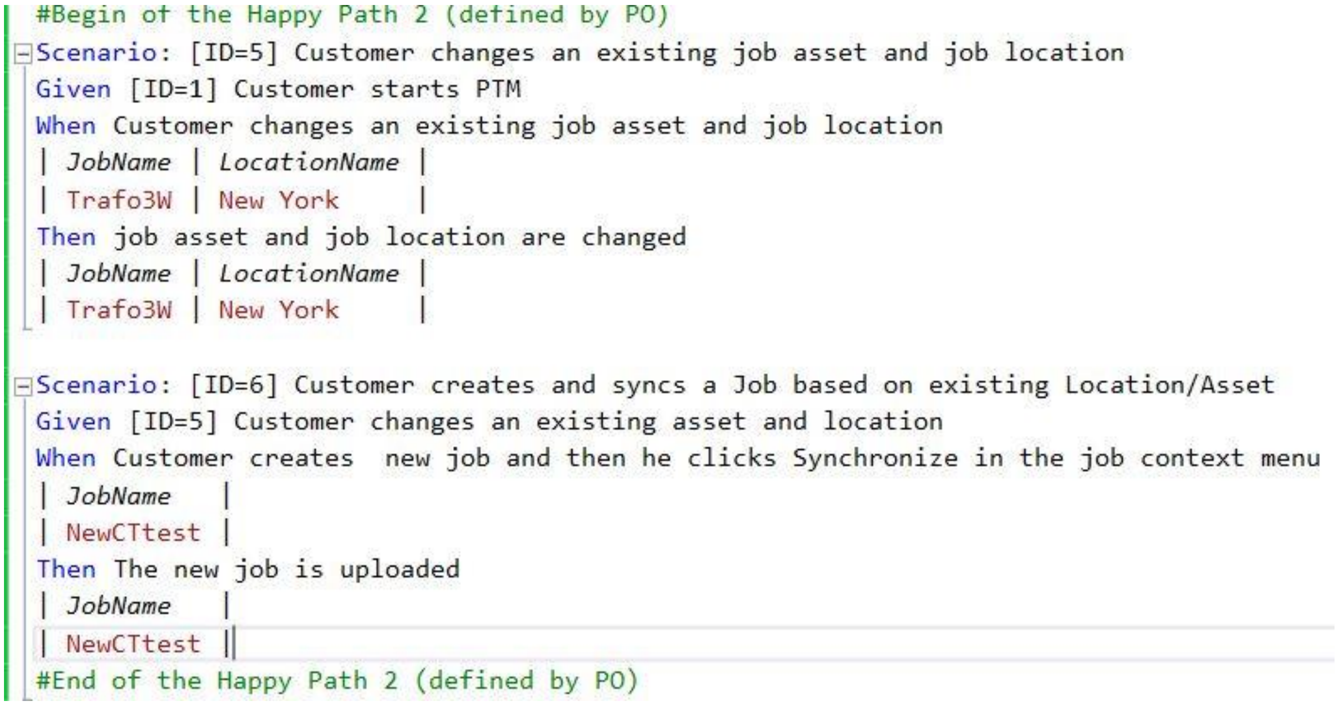
The Process: Implementation

- Selection of the formal notation
  - Requirement 1: it is possible to derive a subset of UML state-charts
  - Requirement 2: contain as much natural language as possible
  - Requirement 3: minimize the number of new tools to introduce the process
- Industrial and academic literature research
  - New domain specific language (DSL) vs. the existing one
  - Existing DSL: Gherkin (Behaviour Driven Development)
- Transform Gherkin into a subset of UML state-charts
  - First idea in 2008 in an industrial blog [1]
  - First case-study in Scerri [2] 2014 showed promising results

Case Study: General Facts

- Primary Test Manager (PTM) supports the workflow for analysing the condition of transformers
    - Jobs are executed on assets which are assigned to locations
    - System size: approximately 400.000 LOC
- PTM Server Integration
    - Client-server application which allows the synchronization of jobs to a server
    - Feature under test: synchronisation of a single job to a server
- Toolchain
    - *Gherkin* for Visual Studio -> SpecFlow
    - *Gherkin* to state-chart -> dedicated tool (*Text2Mod*)
    - Visualization by *yED* editor
    - Test case generation by *TAI* Framework [3]

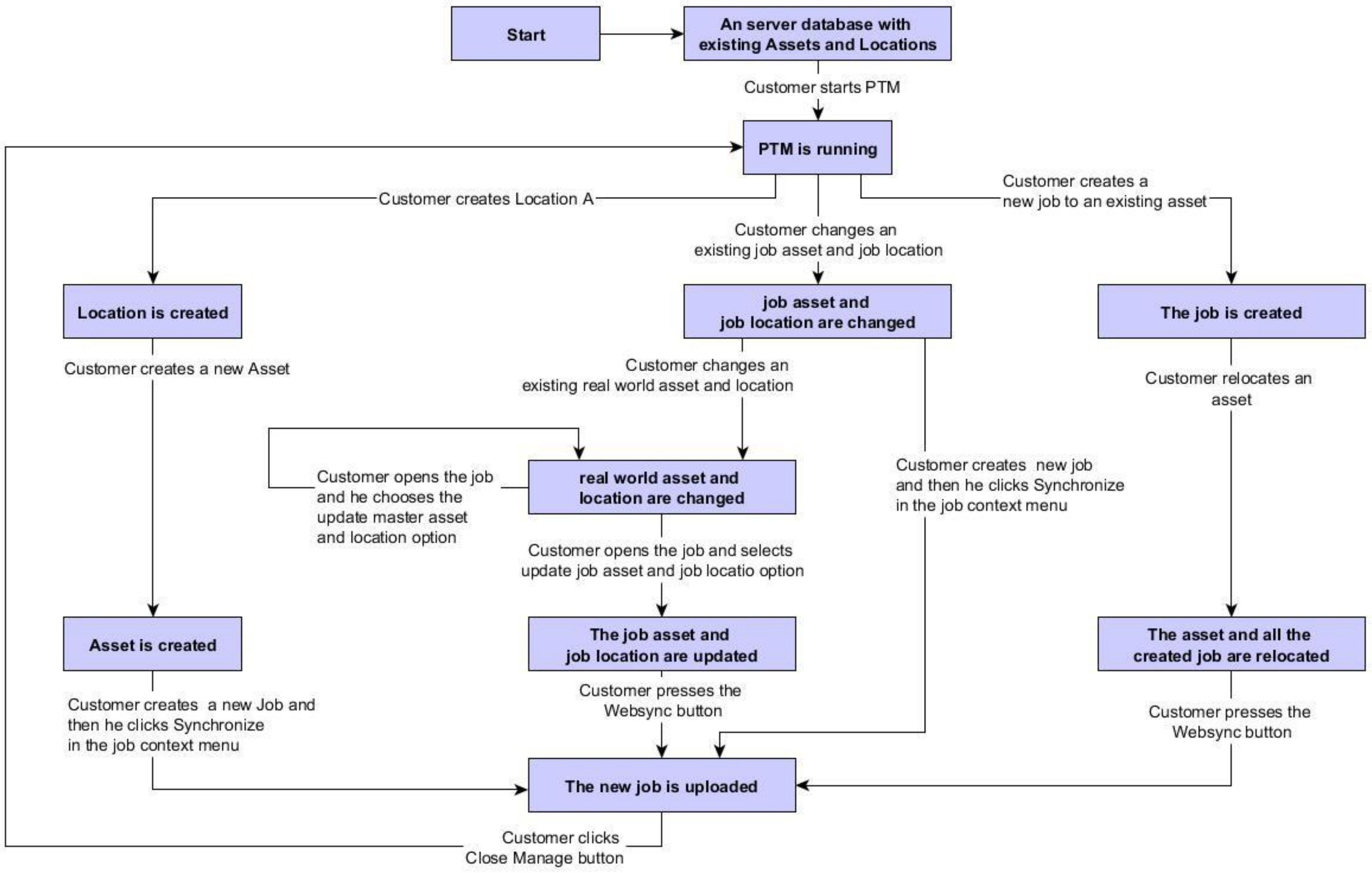
Case Study: Example of Gherkin Path Definition



```
#Begin of the Happy Path 2 (defined by PO)
Scenario: [ID=5] Customer changes an existing job asset and job location
Given [ID=1] Customer starts PTM
When Customer changes an existing job asset and job location
| JobName | LocationName |
| Trafo3W | New York     |
Then job asset and job location are changed
| JobName | LocationName |
| Trafo3W | New York     |

Scenario: [ID=6] Customer creates and syncs a Job based on existing Location/Asset
Given [ID=5] Customer changes an existing asset and location
When Customer creates  new job and then he clicks Synchronize in the job context menu
| JobName   |
| NewCTtest |
Then The new job is uploaded
| JobName   |
| NewCTtest |
#End of the Happy Path 2 (defined by PO)
```

Case Study: Example of Generated Usage Model


Start
An server database with existing Assets and Locations
Customer starts PTM
PTM is running
Customer creates Location A
Customer creates a new job to an existing asset
Customer changes an existing job asset and job location
Location is created
job asset and job location are changed
The job is created
Customer creates a new Asset
Customer changes an existing real world asset and location
Customer relocates an asset
Customer opens the job and he chooses the update master asset and location option
real world asset and location are changed
Customer creates new job and then he clicks Synchronize in the job context menu
Customer opens the job and selects update job asset and job locatio option
Asset is created
The job asset and job location are updated
The asset and all the created job are relocated
Customer creates a new Job and then he clicks Synchronize in the job context menu
Customer presses the Websync button
Customer presses the Websync button
The new job is uploaded
Customer clicks Close Manage button

Case Study: Evaluation

- Approach presented to the product owner (PO)
  - 7 minutes needed to explain the notation
  - 5 minutes needed by the PO to define one happy path and to review the partial model
- SWQA needed approx. 90 minutes to define the set of alternative paths
- The final usage model review took around 30 minutes
- The final usage model consisted of 11 states and 14 transitions
- General feedback
  - PO could imagine using the notation
  - Usability improvements needed

Possible Limitations and Future Work

- Automatically generated model needs to be enhanced by the SWQA to fully comply with UML
- PO has a different perspective on the requirements than SWQA thus he possibly might omit certain aspects while defining the set of happy paths
- Future work 1: evaluate the approach in other PTM teams
- Future work 2: explore the possibility of reusing certain scenarios across different teams
- Future work 3: increase the usability of the approach

Thank You for Your Attention

Literature

[1] The Truth About BDD, CleanCoder.

https://sites.google.com/site/unclebobconsultingllc/the-truth-about-bdd

[2] C. Colombo, M. Micallef, M. Scerri, "Verifying Web Applications: From Business Level Specifications to Automated Model-Based Testing" EPTCS 141 9th Workshop on Model-Based Testing, 2014, Grenoble, France

[3] Winder, M. Automation of regression testing based on usage models.

Master's thesis, Technical University of Graz, 2012