Towards Safety Standard Compliance of IoT Software Systems Using Modelling and Verification with DCR Graphs

Anastasia Orishchenko

Department of Informatics, University of Oslo

June 2021

Supervisors: Christian Johansen, Olaf Owe

Motivation

Boeing discovers new software problem in 737 Max

The US aerospace giant said it is "keeping our customers and suppliers informed" about the new software issue. Aviation regulators have grounded the 737 Max across the globe after two deadly crashes.

KILLED BY A MACHINE: THE THERAC-25

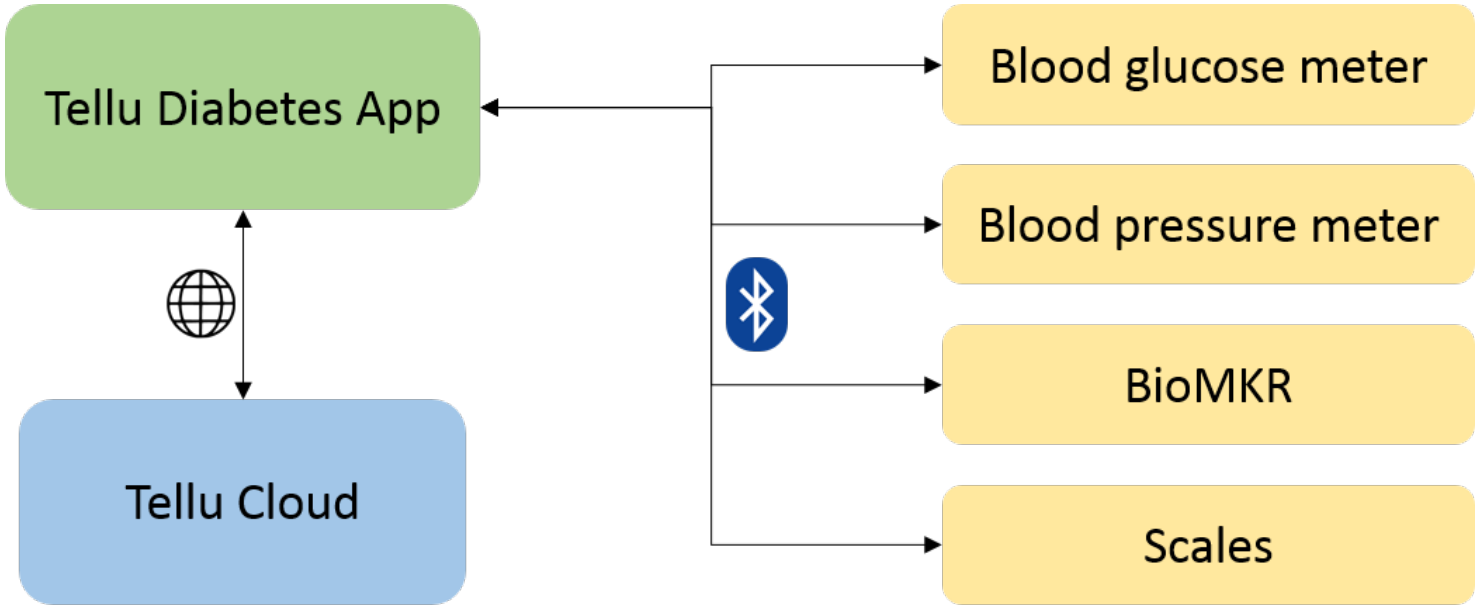
Nissan Recalls Nearly 1 Million Cars for Air Bag Software Fix

Motivation

- Safety standards
- Formal modelling and verification
- Modern systems are modified rapidly
- Modelling and verification of the software development process

Tellu Diabetes App

- Mobile application
- Part of a critical IoT system
- Report health measurements



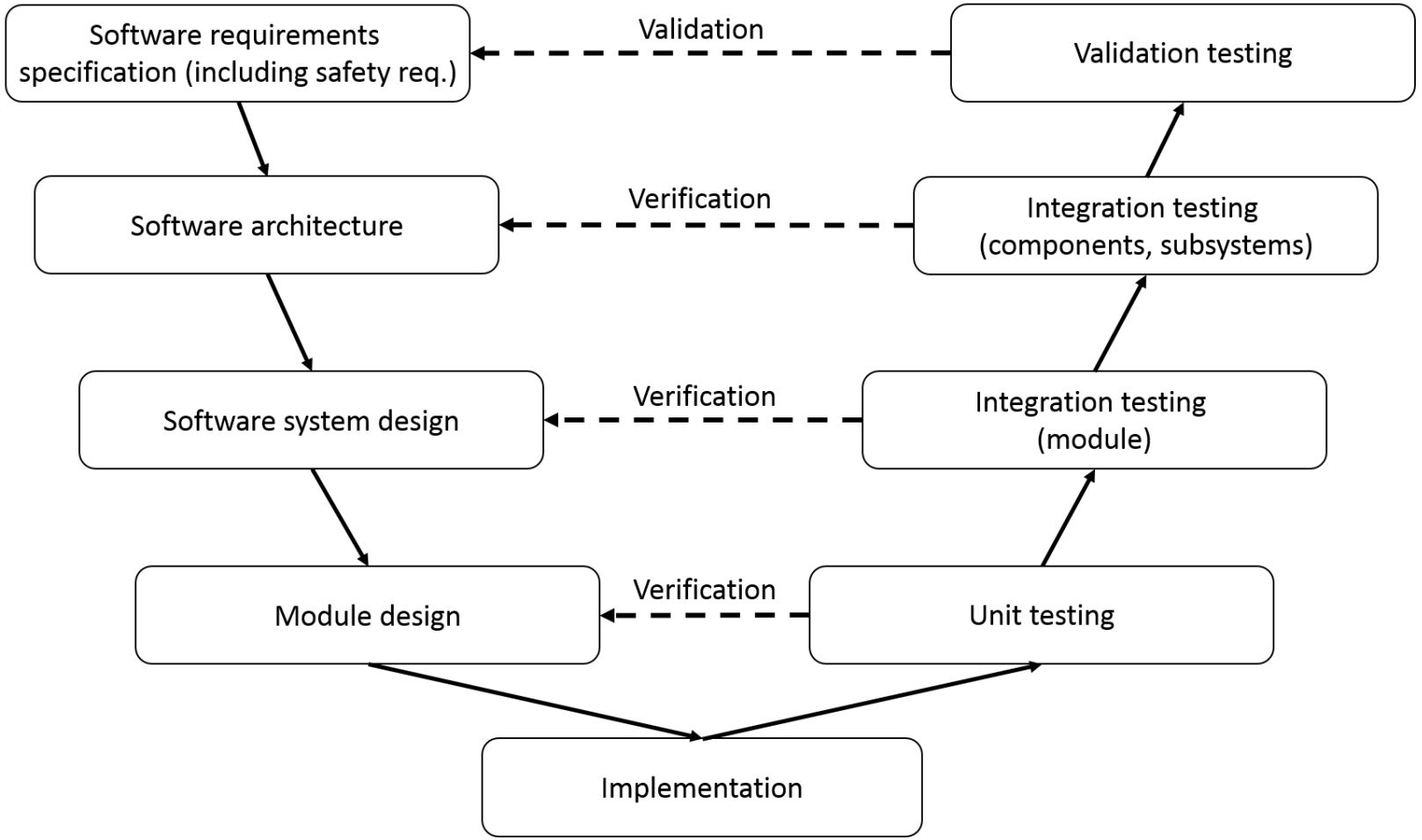
Objectives

1. Study how modelling and verification of the process of introducing new changes to the code can contribute to satisfying safety standard requirements.
2. Investigate how well Dynamic Condition Response Graphs (DCR Graphs) can be applied to modelling a real Internet of Things (IoT) system as a part of the software development process.

Safety standards

- Ensure the safety of the systems developed
- Reduce the risk of error occurence
- Safety integrity level (SIL)
- Requirements related to hardware and software components
- IEC 61508 - Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems

Safety standards



Safety standards

Technique/measure	SIL1	SIL2	SIL3	SIL4
Computer-aided specification tools	Recommended	Recommended	Highly recommended	Highly recommended
Semi-formal methods	Recommended	Recommended	Highly recommended	Highly recommended
Formal methods	—	Recommended	Recommended	Highly recommended

Table 2.1: Requirements involving software specification

Safety standards

Technique/measure	SIL1	SIL2	SIL3	SIL4
Computer-aided design tools	Recommended	Recommended	Highly recommended	Highly recommended
Semi-formal methods	Recommended	Highly recommended	Highly recommended	Highly recommended
Formal methods	—	Recommended	Recommended	Highly recommended
Modular approach	Highly recommended	Highly recommended	Highly recommended	Highly recommended
Structured programming	Highly recommended	Highly recommended	Highly recommended	Highly recommended
Use of trusted/verified software modules (if available)	Recommended	Highly recommended	Highly recommended	Highly recommended
Design and coding standards	Recommended	Highly recommended	Highly recommended	Highly recommended
Defensive programming	—	Recommended	Highly recommended	Highly recommended

Table 2.2: Requirements for software design and implementation

Safety standards

Technique/measure	SIL1	SIL2	SIL3	SIL4
Formal proof	—	Recommended	Recommended	Highly recommended
Static analysis	Recommended	Highly recommended	Highly recommended	Highly recommended
Dynamic analysis and testing	Recommended	Highly recommended	Highly recommended	Highly recommended
Software complexity metrics	Recommended	Recommended	Recommended	Recommended

Table 2.3: Requirements for software verification

DCR Graphs

- Modelling of business processes
- Alternative to existing notations (BPMN, flow charts, swim lane diagrams)
- Declarative, event-based model
- Developed in collaboration between IT University of Copenhagen and Exformatics A/S
- Academic publications and industrial case studies (healthcare, railway sector)

DCR Graphs: Structure

- Directed graph
  - Nodes = events/activities
  - Edges = relations between events
- Events/activities
  - Name, description
  - Role(s)
  - Included, pending, executed
  - Relations to itself or other activities
- Grouping of activities

DCR Graphs: Relations

- Condition, milestone and pre-condition

- Response and No-Response

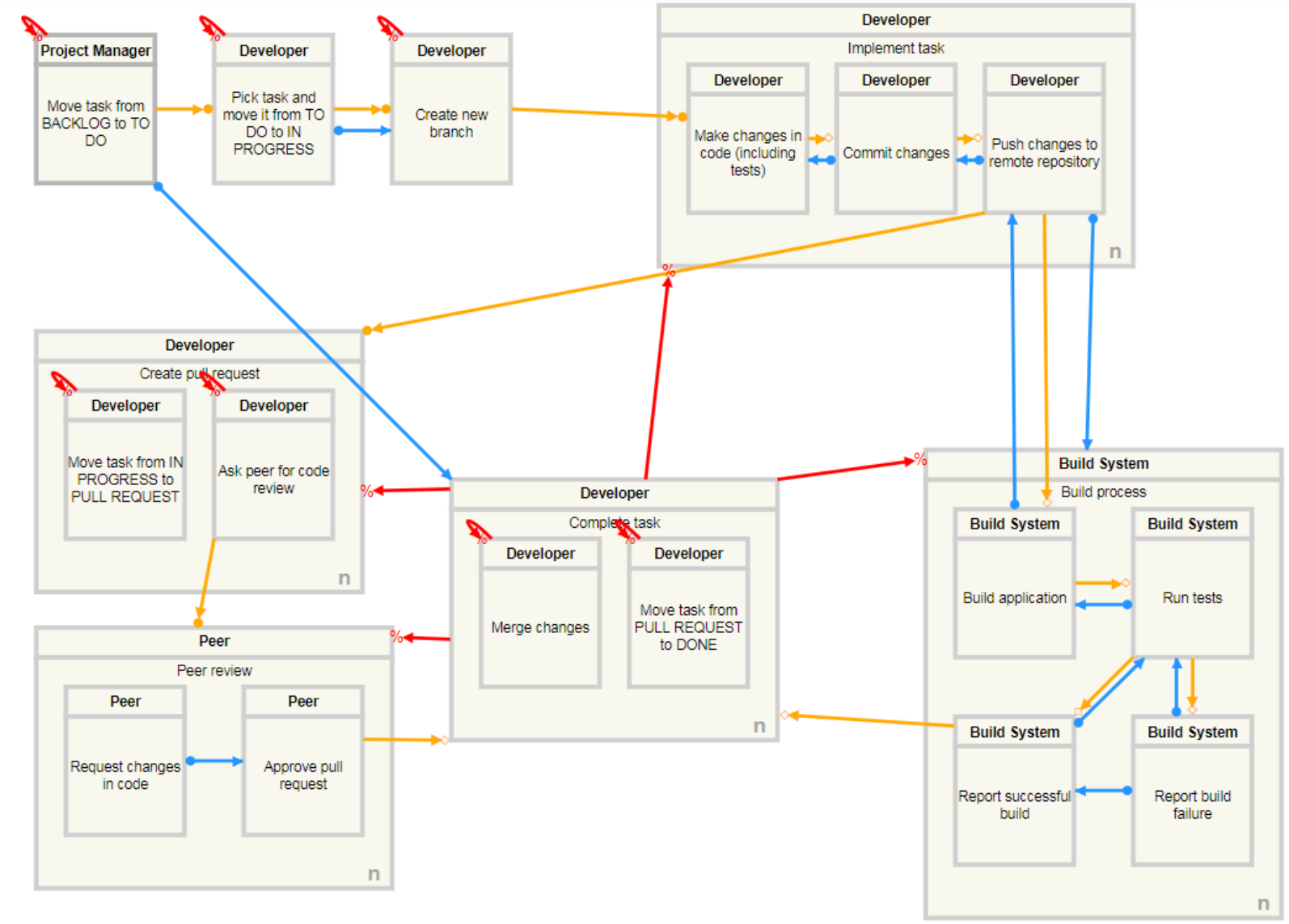
- Include

- Exclude

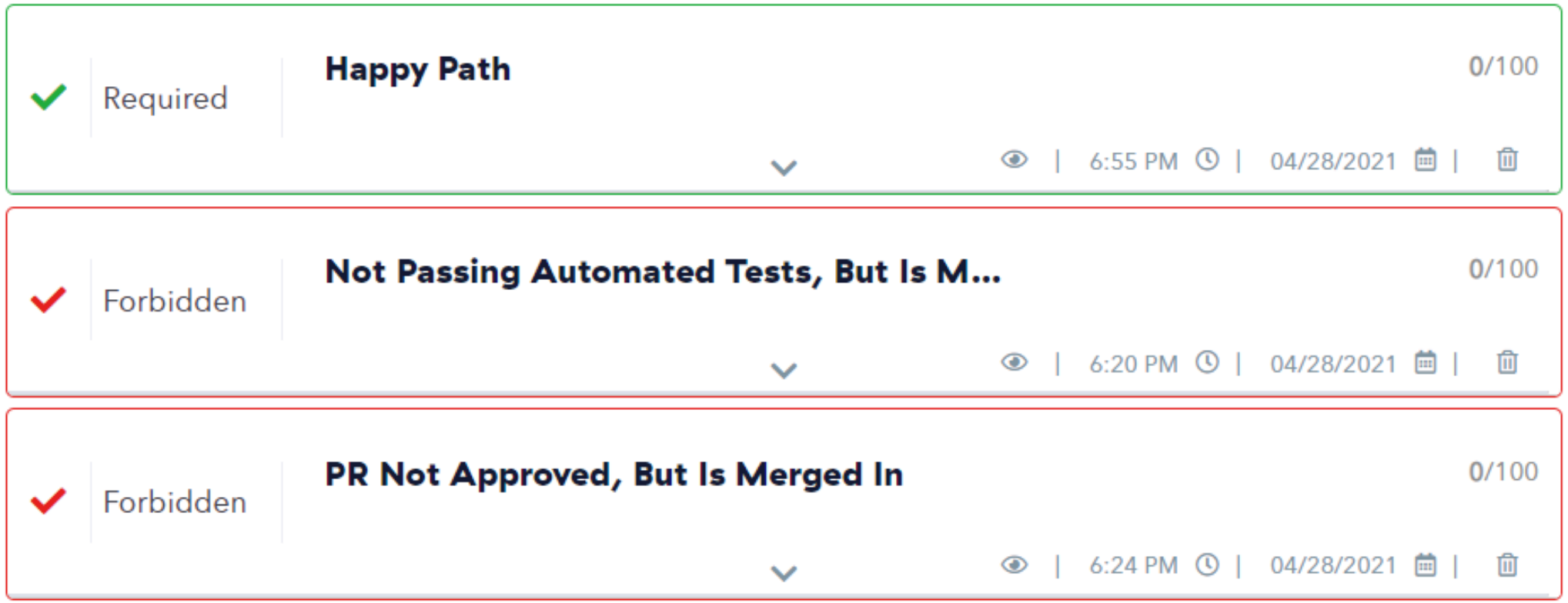
Methodology

- DCR process methodology
  - Define activities
  - Define roles
  - Define rules
- DCR Tool
- Verification of model behaviour
  - DCR Swimlane Editor
  - Scenario Search application
  - Dead-end Analyzer application

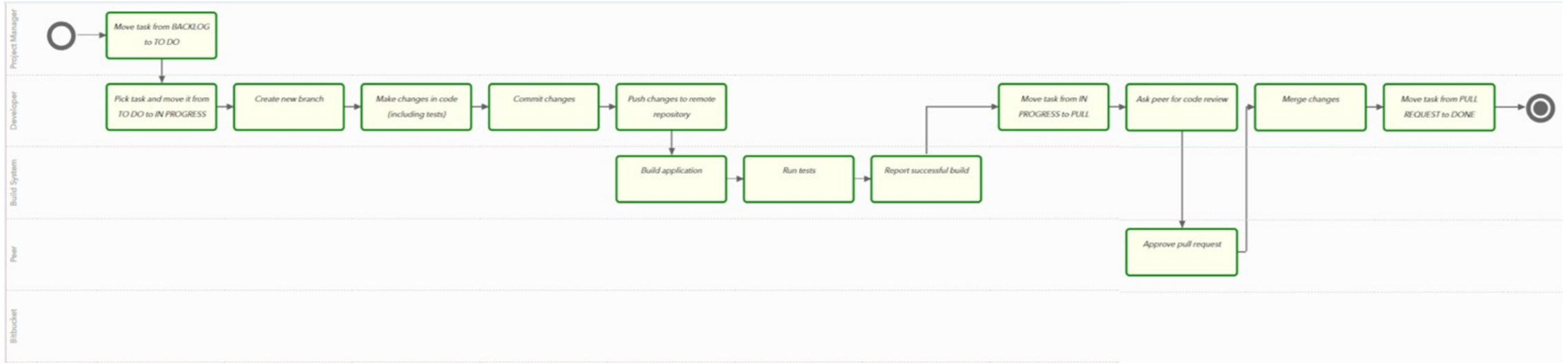
Model of the task implementation process

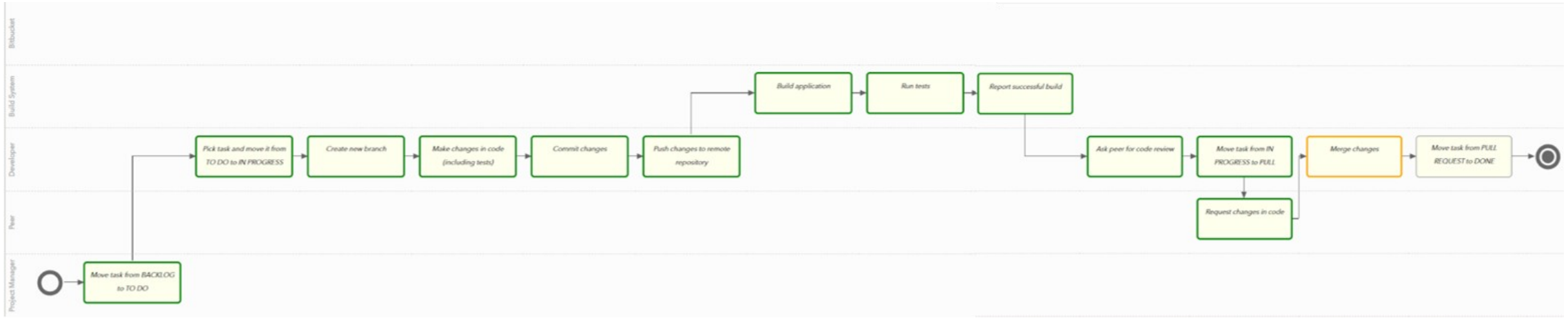


Verification of the task implementation process

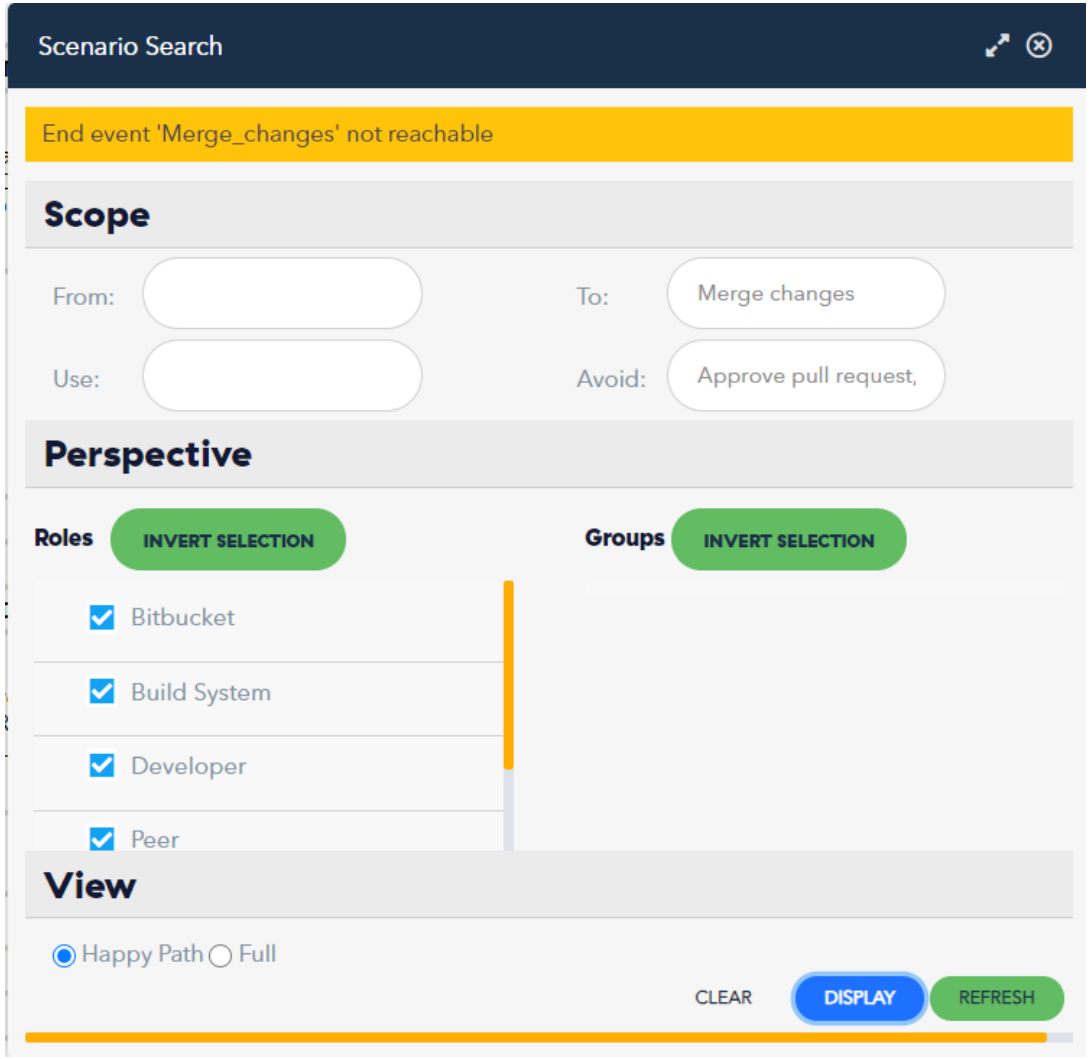


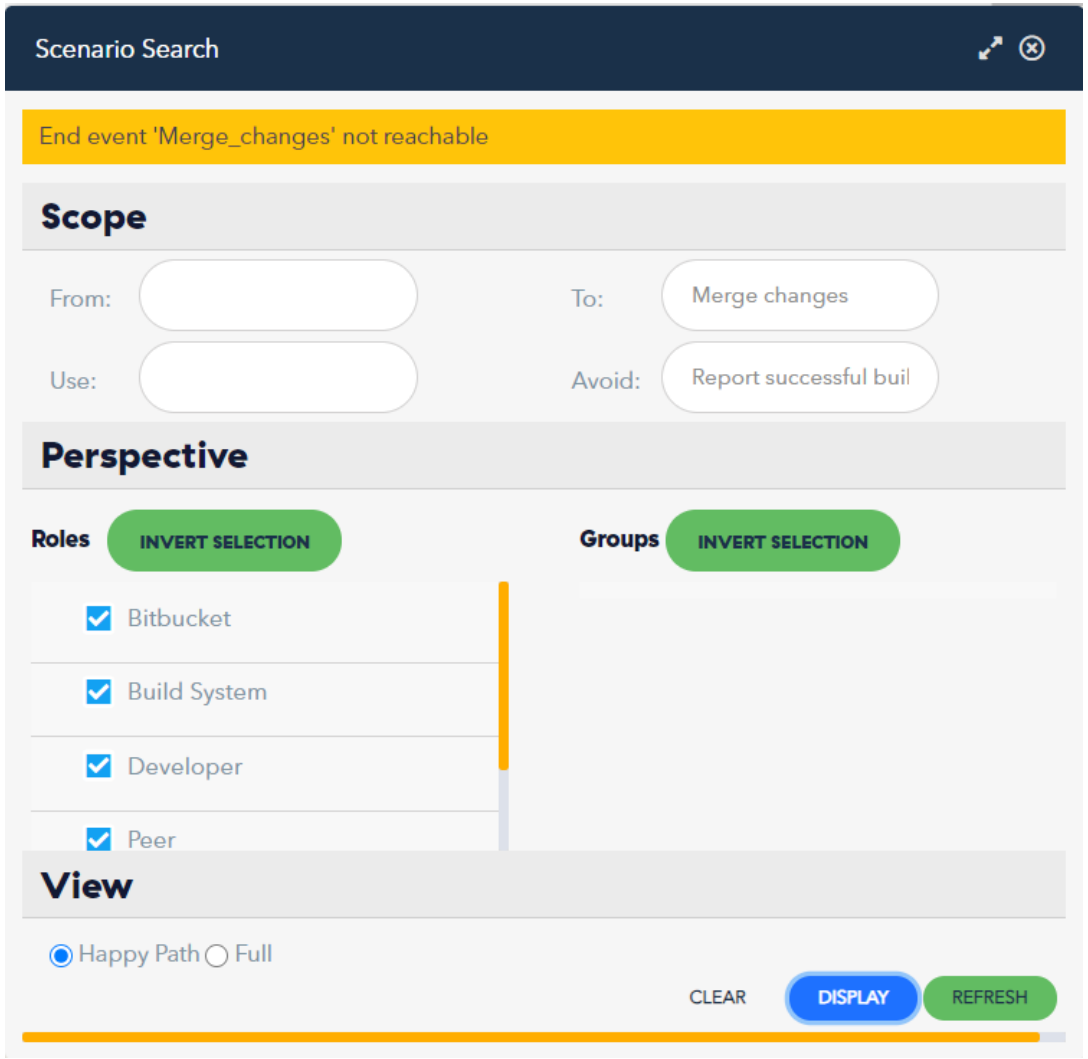
Verification of the task implementation process



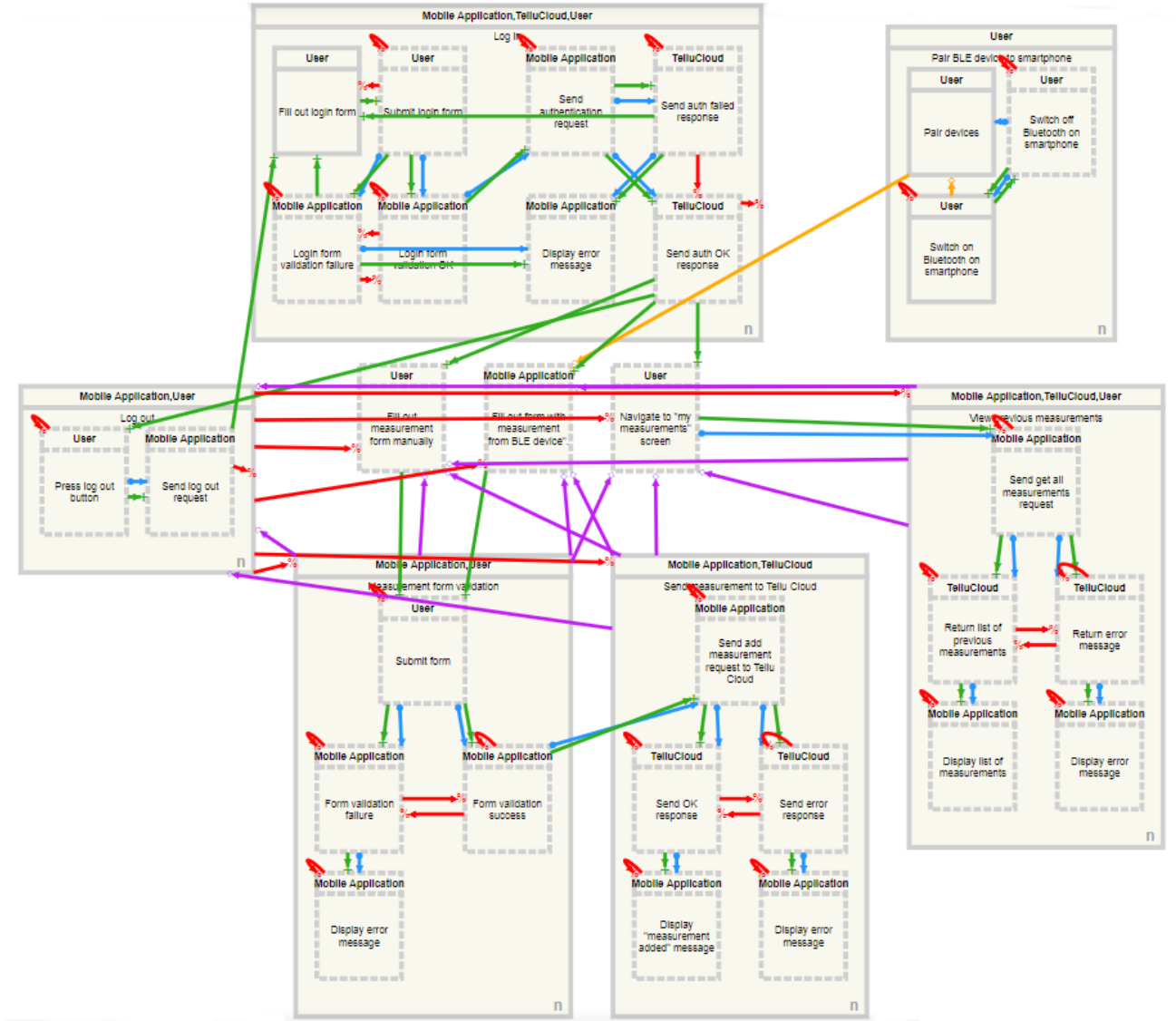


Verification of the task implementation process

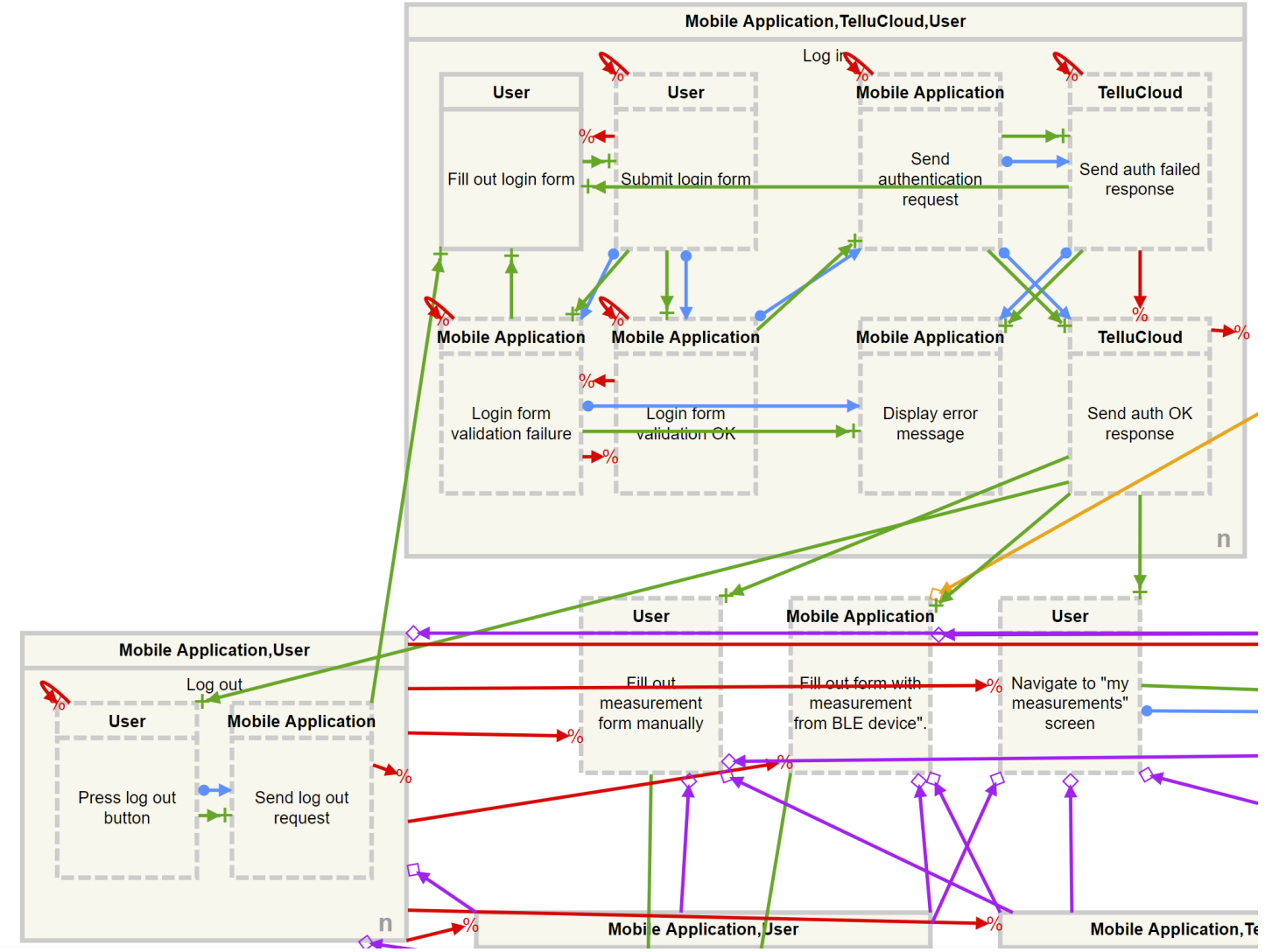




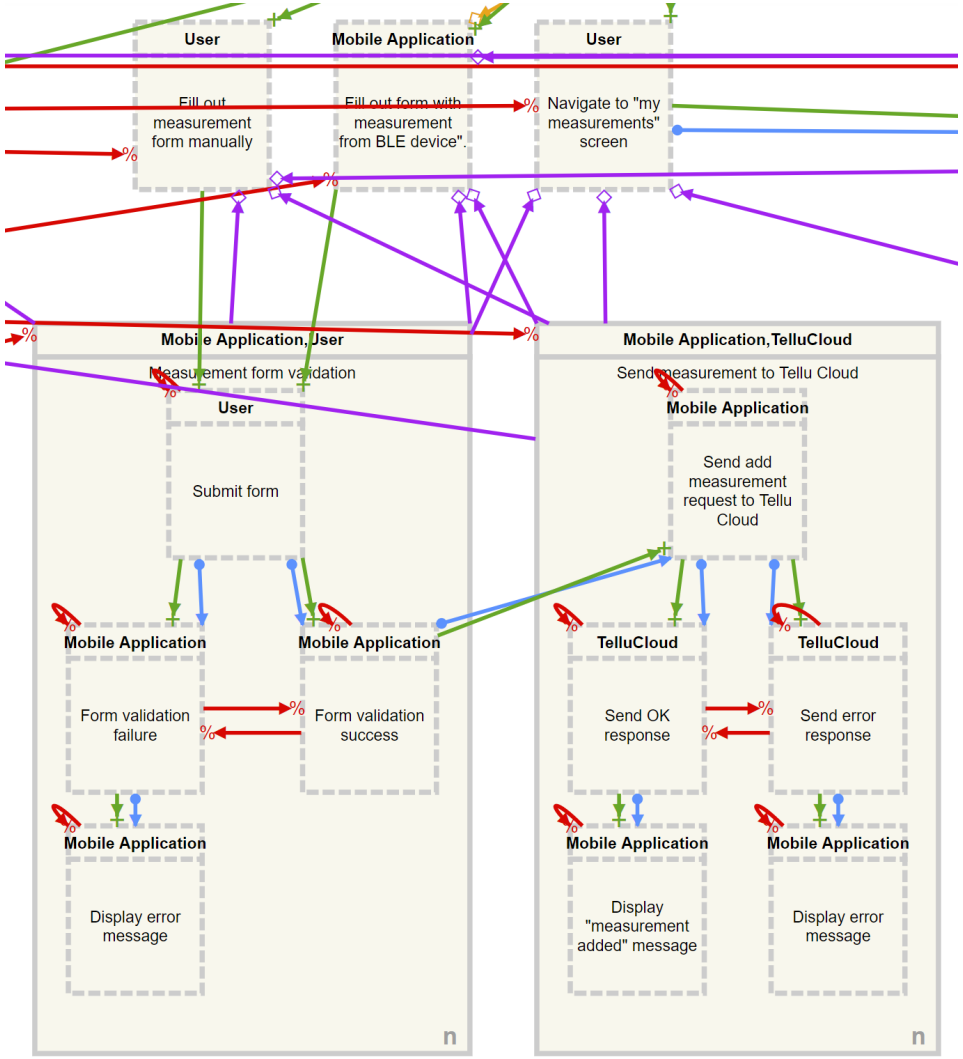
Model of Tellu Diabetes App functionality



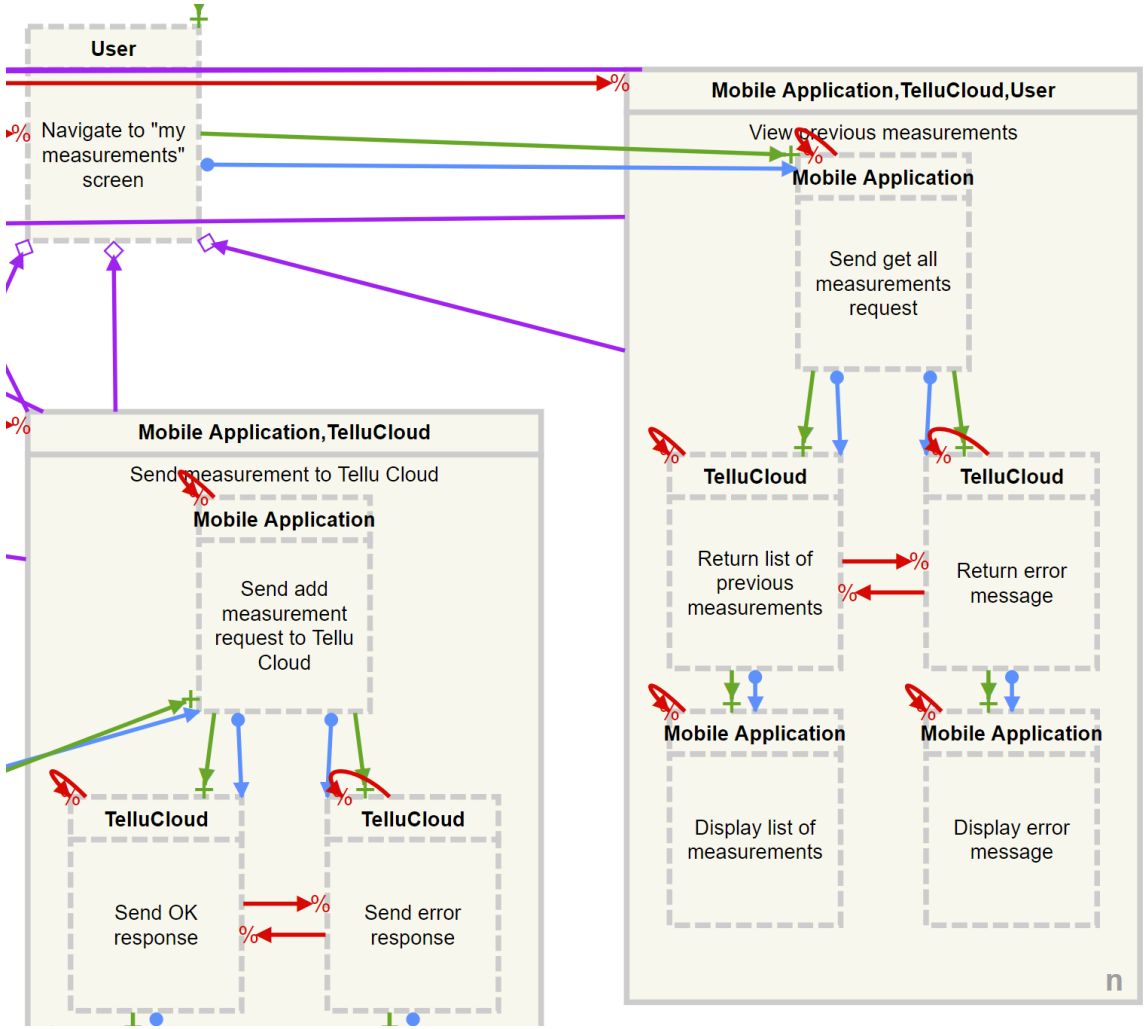
Model of Tellu Diabetes App functionality - Log in and log out



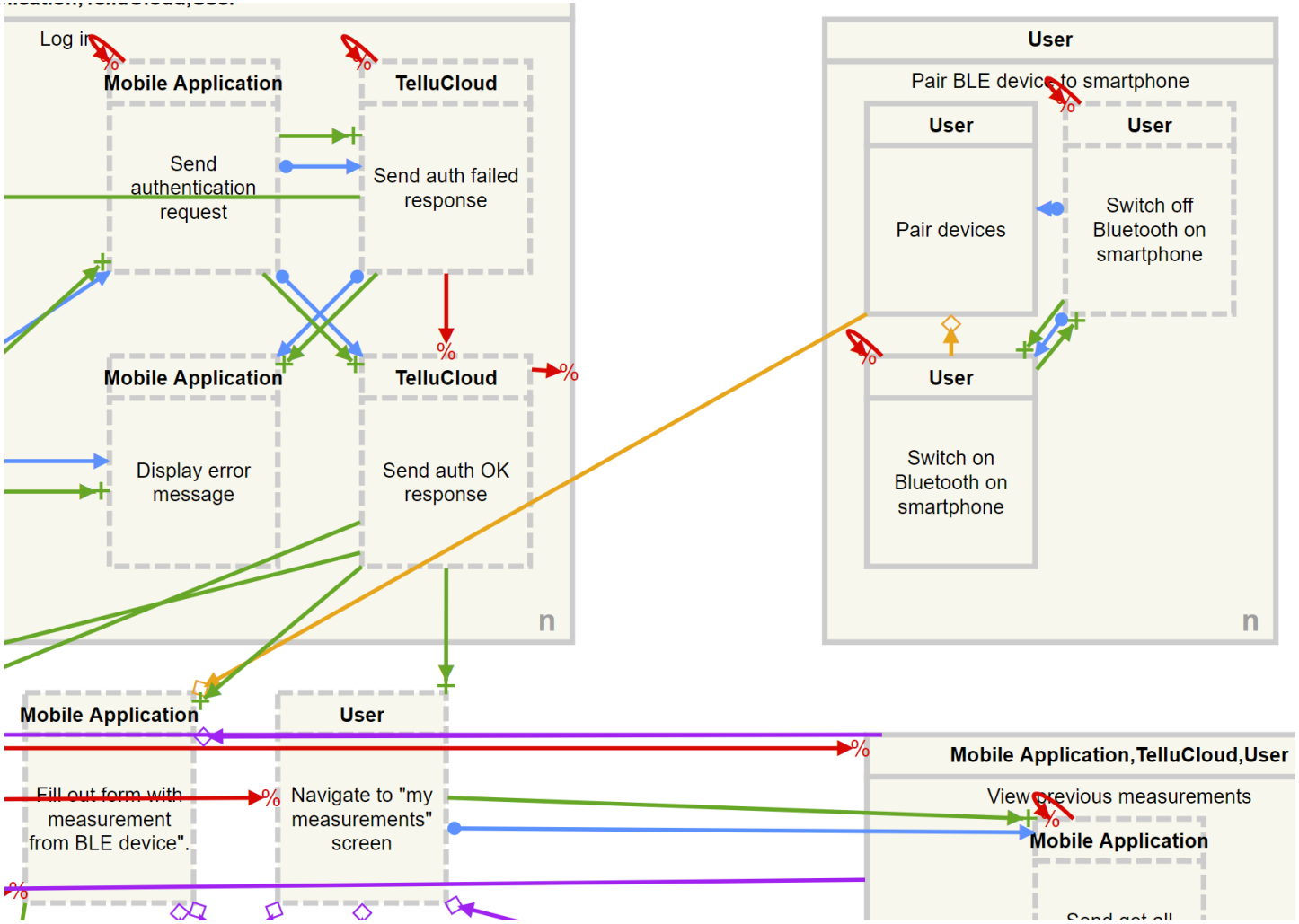
Model of Tellu Diabetes App functionality - Submitting measurements



Model of Tellu Diabetes App functionality - View previous measurements



Model of Tellu Diabetes App functionality - Switching Bluetooth on and off



Discussion: Towards satisfying requirements for software development process

- Peer review
- Build process including running automated tests
- These steps are executed for each modification introduced to the code
- Weakness: model does not necessarily reflect the actual process

Technique/measure	SIL1	SIL2	SIL3	SIL4
Formal proof	—	Recommended	Recommended	Highly recommended
Static analysis	Recommended	Highly recommended	Highly recommended	Highly recommended
Dynamic analysis and testing	Recommended	Highly recommended	Highly recommended	Highly recommended
Software complexity metrics	Recommended	Recommended	Recommended	Recommended

Table 2.3: Requirements for software verification

Discussion: DCR Graphs for modelling of Internet of Things system

- Operations in a system are similar to activities in a business process
- Operations → activities with roles
- Dependencies between operations → relations
- No end goal
- Representation of current state
- Easily extendable
- Complexity
- Graphical notation
- Contribute to satisfy safety standard requirements

Future work

- Lower abstraction level in the model of Tellu Diabetes App functionality
  - Extend the model further with more code-specific activities
- Investigate how the models of the task implementation process and the functionalities of the application can be joined
  - Add more aspects of software development process

Conclusion

- Modelled task implementation process and Tellu Diabetes App functionality as DCR Graphs
- Verified some of the desired properties
- Theoretical knowledge and practical experience

Thank you for attention!