Improving Coordination and Collaboration in Connected and Automated Vehicle Development Projects

Using Model Based Process Design

Avi Shaked
&
Yoram Reich

Systems Engineering Research Initiative
Tel Aviv University
Improving Coordination and Collaboration in Connected and Automated Vehicle Development Projects

Using Model Based Process Design

Avi Shaked
&
Yoram Reich

Systems Engineering Research Initiative
Tel Aviv University
Billions of $$$ at stake!

Multiple players!
Improving Coordination and Collaboration in Connected and Automated Vehicle Development Projects

Why?
“Plans are nothing, Planning is everything”

- Dwight D. Eisenhower
Improving Coordination and Collaboration in Connected and Automated Vehicle Development Projects
Development Processes Should Be Properly & Continuously Planned

1. Establish goals
2. Identify prerequisites
3. Allow manageability
4. Detect change/anomaly/variation/deviation

Not a good human interface for planning!
Introducing PROVE

PROVE – Process Oriented Viewpoint Engineering

A Development Process Design Framework

Notation & Method
PROVE Notation

Activity 1

Artifact A::InputState → Activity 1-A → Artifact A::OutputState

Artifact B::Expected → Artifact C::Created
PROVE Method

Project
PROVE Method

I: Desired Goals

Project

System::Validated

DesignDescription::Released

System::Certified
PROVE Method

II: Inputs

- CustomerRequirements::Released
- InterfacingSystemICD::Delivered
- CertificationStandard::Delivered

Project

- System::Validated
- DesignDescription::Released
- System::Certified
PROVE Method

III: Activities

- TestRange::Functional
- CustomerRequirements::Released
- CodingStandard::Initial
- InterfacingSystem::Delivered
- CertificationStandard::Delivered
- DesignDescription::Released
- System::Validated
- DesignDescription::Released
- System::Certified
PROVE Method

IV: Recursive planning

For additional method steps see our SoSE 2018 paper (DOI: 10.1109/SYSOSE.2018.8428767)!
Development Processes Should Be Properly & Continuously Planned

1. Establish goals
2. Identify prerequisites
3. Allow manageability
4. Detect change/anomaly/variation/deviation
1. Use PROVE to Establish Goals
1. Use PROVE to Establish Goals

In different hierarchies!
2. Use PROVE to Specify Prerequisites
3. Use PROVE to Manage Engineering

Vehicle Development

Computer S/W Dev. (subcontractor A)

Computer Software Design Process

Software Architecture

Low Level Requirements

Derived Low Level Requirements

Vehicle Certification

Certification Package

Approved
4. Use PROVE to Detect Change
PROVE is Beneficial for System Development Efforts

• Approachable implementation
• Process orientation
  – Allows establishing scopes & policy
  – Promotes multi-level process consistency & completeness
• Communicable, structured model
  – Promotes better process coordination
  – Facilitates pattern mining & reuse

A disciplined, coordinated development process

Detailed in our SAE Technical Paper!