S-Scrum

Towards Applying a Safety Analysis and Verification Method based on STPA to Agile Software Development

PhD student: Yang Wang
Supervisor: Prof. Dr. Stefan Wagner

University of Stuttgart/ISTE/SE
14.05.2016
Agenda

- Problems
- Research Question
- Overview S-Scrum

Background
- STPA for safety analysis
- Model Checking for safety verification

S-Scrum
- Safe Scrum
- STPA in S-Scrum
- Model Checking in S-Scrum

Example
- Airbag systems

© Copyright Yang Wang @ University of Stuttgart, Institute of Software Technology, Software Engineering Group, 2016
Problems


Problems

Requirement
Design
Implementation
Verification
Maintenance

BDUP
Big Design Up Front

Safety analysis:
FMEA, FTA

IEC 61508
ISO 26262
IEC 61508
EN 50129
ISO 13849
EN 50128
EN 50129
ISO 13849
ISO 25119
EN 50128
EN 50129

http://www.bosch-engineering.de/en/de/dienstleistungen/funktionale_sicherheit/funktionale_sicherheit_1.html

© Copyright Yang Wang @ University of Stuttgart, Institute of Software Technology, Software Engineering Group, 2016
Problems

No BDUP
Big Design Up Front

© Copyright Yang Wang @ University of Stuttgart, Institute of Software Technology, Software Engineering Group, 2016
Without a stable architecture, how could Scrum perform safety assurance for safety-critical systems?
Overview

By combining formal verification method – Model Checking

By integrating a safety analysis method - STPA

S-Scrum

A Scrum development process for safety-critical systems

Based on Safe Scrum software development process

© Copyright Yang Wang @ University of Stuttgart, Institute of Software Technology, Software Engineering Group, 2016
Agenda

- Problems

- Overview S-Scrum

- Background
  - STPA for safety analysis
  - Model Checking for safety verification

- S-Scrum
  - Safe Scrum
  - STPA in S-Scrum
  - Model Checking in S-Scrum

- Example
  - Airbag systems
A new hazard analysis approach, proposed by Nancy Leveson, 2003

Based on STAMP (Systems–Theoretic Accident and Process) causality model, which includes new causal factors.

Safety-Guided Design

1. STAMP Model

2. STPA hazard analysis

How do we find inadequate control in a design?

Accidents are caused by inadequate control

Design Decisions

3. Hazard Analysis (STPA)
Background – STPA (Systems-Theoretic Process Analysis)

1. STAMP Model
   - Identify accidents and hazards
   - Construct the control structure
   - Step 1: Identify unsafe control actions
   - Step 2: Identify causal factors and control flaws

2. STPA hazard analysis

<table>
<thead>
<tr>
<th>STPA step 1</th>
<th>Not providing causes hazard</th>
<th>Providing causes hazard</th>
<th>Incorrect Timing/Order</th>
<th>Stopped Too Soon/Applied too long</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control Action</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

STAMP Model

Controller

Controlled Process

Actuator: Control Actions

Feedbacks

© Copyright Yang Wang @ University of Stuttgart, Institute of Software Technology, Software Engineering Group, 2016
Background – STPA (Systems-Theoretic Process Analysis)
Agenda

- Problems
- Research Question
- Overview S-Scrum

- Background
  - STPA for safety analysis
  - Model Checking for safety verification

- S-Scrum
  - Safe Scrum
  - STPA in S-Scrum
  - Model Checking in S-Scrum

- Example
  - Airbag systems
Background – Model Checking

Model checking is an automated technique that, given a finite-state model of a system and a formal property, systematically checks whether this property holds for (a given state in) that model.

- Christel Baier and Joost-Pieter Katoen, Principles of model checking

- Formulating the safety requirements using temporal logic (LTL/CTL).

- Modeling the source code as input model (PROMELA).

- Verifying the safety requirements specification on the input model (SPIN).
Agenda

- Background
  - STPA for safety analysis
  - Model Checking for safety verification

- S-Scrum
  - Safe Scrum
  - STPA in S-Scrum
  - Model Checking in S-Scrum

- Example
  - Airbag systems
Safe Scrum

Disadvantages:

- Inside each sprint, there is a lack of safety assurance
- RAMS validation is a lack of agility

© Copyright Yang Wang @ University of Stuttgart, Institute of Software Technology, Software Engineering Group, 2016
S-Scrum
Agenda

- Problems
- Research Question
- Overview S-Scrum

- Background
  - STPA for safety analysis
  - Model Checking for safety verification

- S-S²scrum
  - Safe Scrum
  - STPA in S²-Scrum
  - Model Checking in S²-Scrum

- Example
  - Airbag systems
**System Safety Goal:** During a critical crash, protect the passengers from being injured.

**Accident:** The occupants in the target vehicle are injured.

**Hazard 1:** The airbag is not ignited even though a critical crash occurred.

**Hazard 2:** The airbag is ignited even though no crash at all or only a non-critical crash has occurred.

**Hazard 3:** The airbag is ignited after $T = 45$ms.
**Example of S-Scrum (Airbag systems)**

### Examples of STPA step 1

<table>
<thead>
<tr>
<th>Control Action</th>
<th>Not Provided</th>
<th>Provided</th>
<th>Too Soon/Late</th>
<th>Too Long/Short</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Send fire command</strong></td>
<td>UCA.1: Not sending fire command is hazardous if there was a critical crash. [H.1]</td>
<td>UCA.2: A fire command is needlessly sent to the FET and FASIC, thus causing an unintended deployment of the airbag. [H.2]</td>
<td>UCA.3: The fire command for the airbag in case of a crash is delayed, thus causing the airbag to be ignited too late. [H.1][H.3]</td>
<td>/</td>
</tr>
</tbody>
</table>
Example of S-Scrum (Airbag systems)
Example of S-Scrum (Airbag systems)

<table>
<thead>
<tr>
<th>Hazardous Scenario</th>
<th>Associated Causal Factors Leading to UCA.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Process Model</td>
<td>• ABS does not detect the driver is present, thus does not been enabled.</td>
</tr>
<tr>
<td>• ABS Enabled: [Yes, No]</td>
<td></td>
</tr>
<tr>
<td>• Driver Present: [Yes, No]</td>
<td>• Acceleration threshold is incorrect and allows the vehicle get an abnormal acceleration without sending a signal.</td>
</tr>
<tr>
<td>• Acceleration Speed ≥ Threshold: [Yes, No]</td>
<td></td>
</tr>
<tr>
<td>• Roll Rate ≥ Threshold: [Yes, No]</td>
<td></td>
</tr>
<tr>
<td>(2) Sensors</td>
<td>• Pressure sensor is unable to detect pressure due to road and weather conditions.</td>
</tr>
<tr>
<td>• Acceleration Sensors</td>
<td></td>
</tr>
<tr>
<td>• Pressure Sensors</td>
<td>• Noise is not adequately filtered and a rapidly acceleration is not real-time detected.</td>
</tr>
<tr>
<td>• Roll Rate Sensors</td>
<td></td>
</tr>
</tbody>
</table>
Examples of Safety requirements

<table>
<thead>
<tr>
<th>Related UCAs</th>
<th>Corresponding safety constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCA.1</td>
<td>SSR.1: The airbag system control unit should provide signal when there was a critical crash.</td>
</tr>
<tr>
<td>UCA.2</td>
<td>SSR.2: The airbag system control unit should not send fire command when there was no crash or non-critical crash.</td>
</tr>
<tr>
<td>UCA.3</td>
<td>SSR.3: The airbag system control unit must send the fire command $T \leq 45$ms, when there was a critical crash.</td>
</tr>
</tbody>
</table>
**Example of S-Scrum (Airbag systems)**

*Examples of Safety requirements in LTL*

SSR.1 = []((acceleration >= safe acceleration threshold & & pressure intensity >= safe pressure threshold & & driver state == present) -> (send fire command))

SSR.2 = []((acceleration <= safe acceleration threshold | | pressure intensity <= safe pressure threshold | | driver state == present) - > !(send fire command))

![Diagram showing the verification process with model checker, source code, safety product backlog, and verification result.]
Conclusion and Future Work

**Conclusion**

- Find a comprehensive method to solve safety related problems in an Agile development process.
- A balance point between Safe Scrum and basic Scrum

**Future Work**

- Verification and validation of S-Scrum
S-Scrum

Towards Applying a Safety Analysis and Verification Method based on STPA to Agile Software Development

PhD student: Yang Wang
Supervisor: Prof. Dr. Stefan Wagner

University of Stuttgart/ISTE/SE
14.05.2016