

# BUILDING TEST SUITES FROM TEST RECORDINGS OF WEB APPLICATIONS

Martin Filipsky, Miroslav Bures, Ivan Jelinek  
Czech Technical University in Prague

WWW/Internet 2012

Madrid 18-21 October 2012

# Introduction

# Introduction

---

## Automated testing of web applications

### Common expectations:

- ★ Testing is expensive, automate it !

### Current problems:

- ★ Automated test scripts are platform dependent
- ★ Application changes, automated test scripts getting obsolete
- ★ Maintenance costs

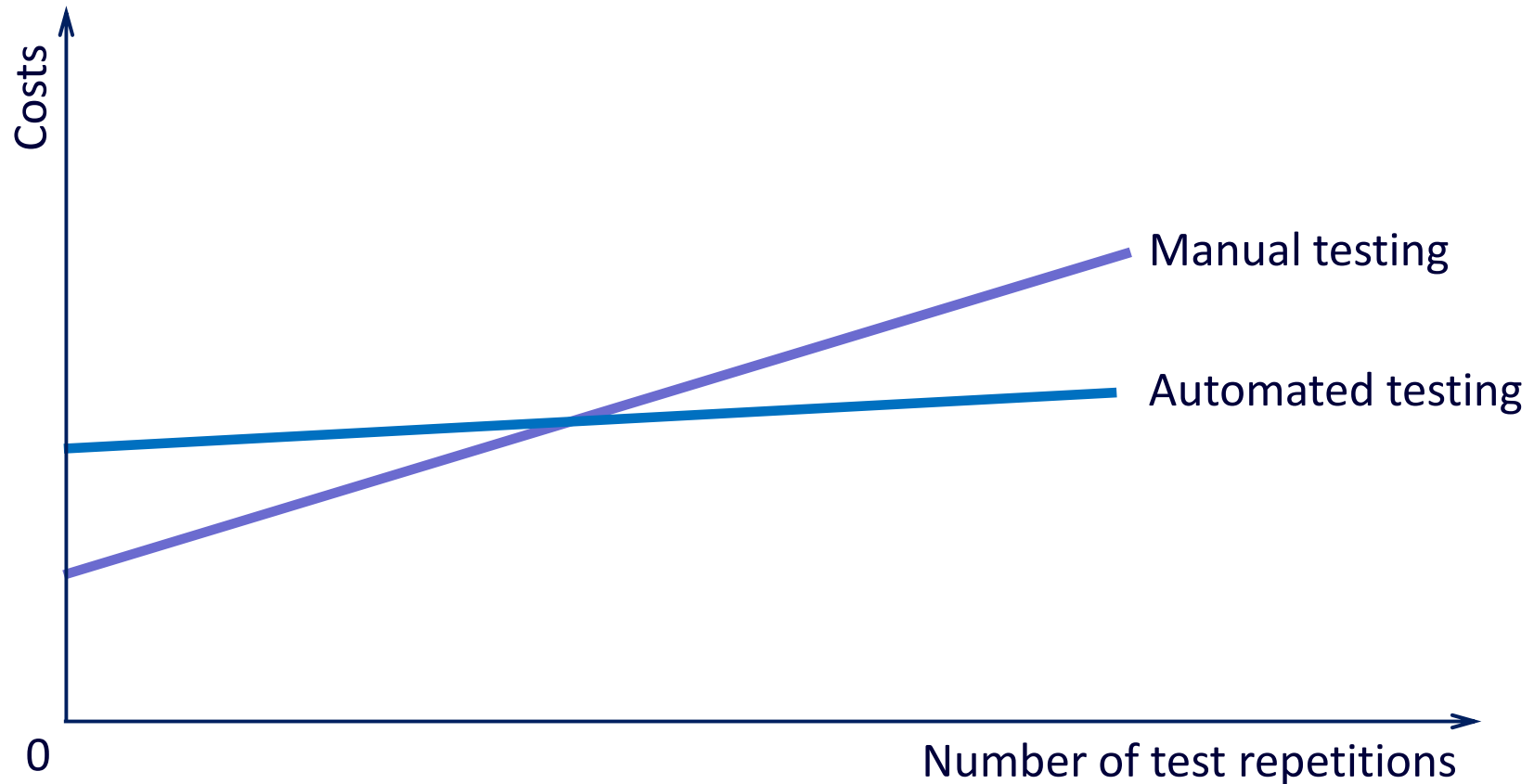
→ Automated test scripts currently being used mainly for regression, smoke and performance testing



# Economics of automation (1/2)

---

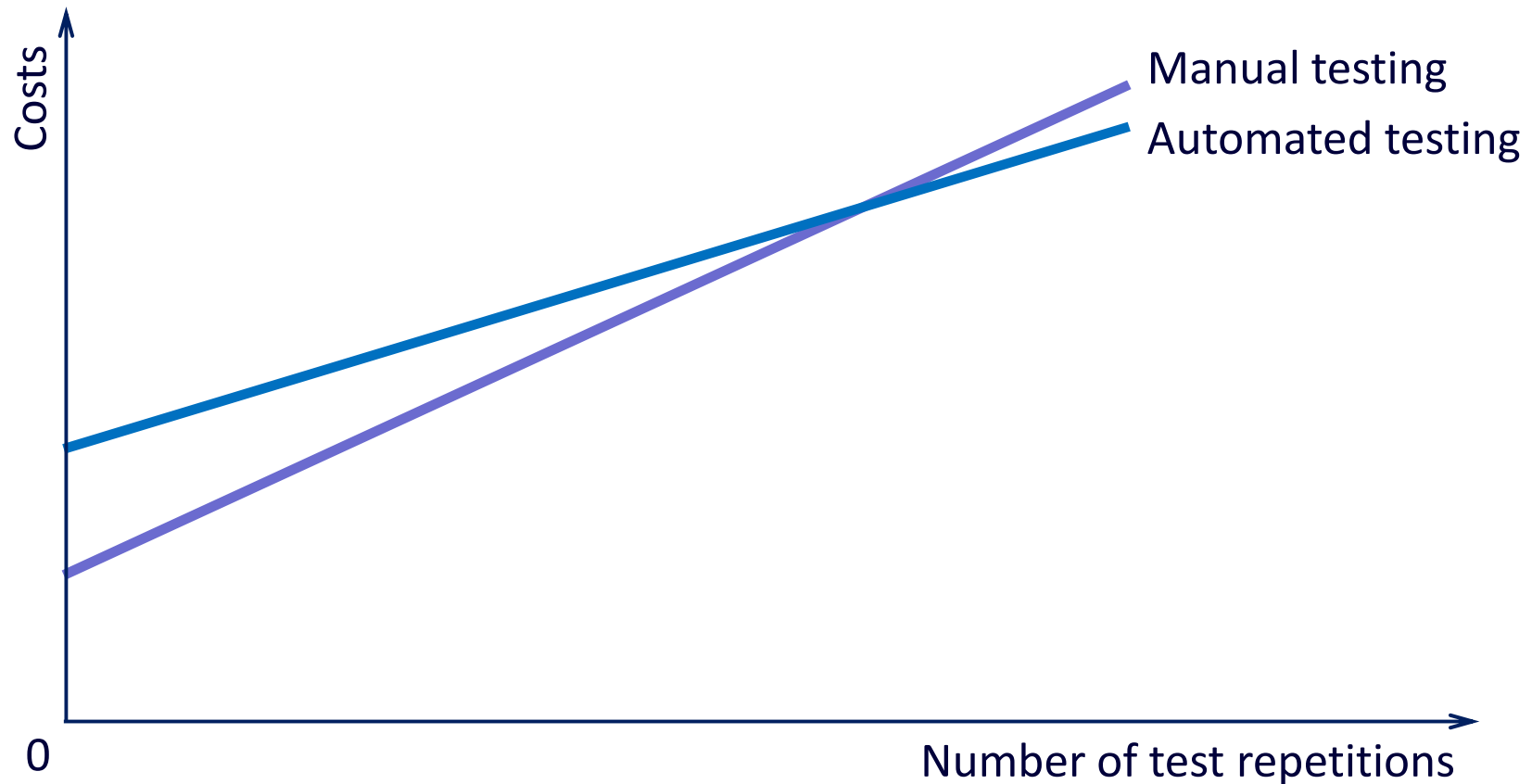
When application under test doesn't change:



# Economics of automation (2/2)

---

But when application under test is changing:



# Current solutions

---

For test automation, several approaches are currently used:

| Type  | Scripts preparation time  | Scripts maintenance       |
|---|---------------------------|---------------------------|
| Plain recording of test cases   | low                       | high                      |
| Test scripting using a programming language   | high                      | medium                    |
| Framework <ul style="list-style-type: none"><li>• Tests driven by data</li><li>• Tests driven by keywords</li><li>• Tests driven by a model</li></ul> | medium                    | medium                    |
| Hybrid approaches   | Can decrease the overhead | Can decrease the overhead |
| <b>What current praxis would need is</b>  | medium                    | low                       |



# **Our research**

# Aim of the research

---

- ★ Propose a solution for automated testing, solving the previous problems:

- Ensuring easier maintenance (prevents scripts ageing)
- Platform independent
- Allowing faster production of automated scripts

- ★ Use the best practices from the previous concepts
- ★ Employ our practical experience from the real projects





# Principle of the solution

---

## Design:

### ★ Formal model (Meta model) describing:

- Requirements on automated tests
- An application under test
- Test scripts to record test cases

### ★ Smart structure of test cases

- Common parts of the recorded tests broken down into reusable pieces for an easier and cheaper maintenance

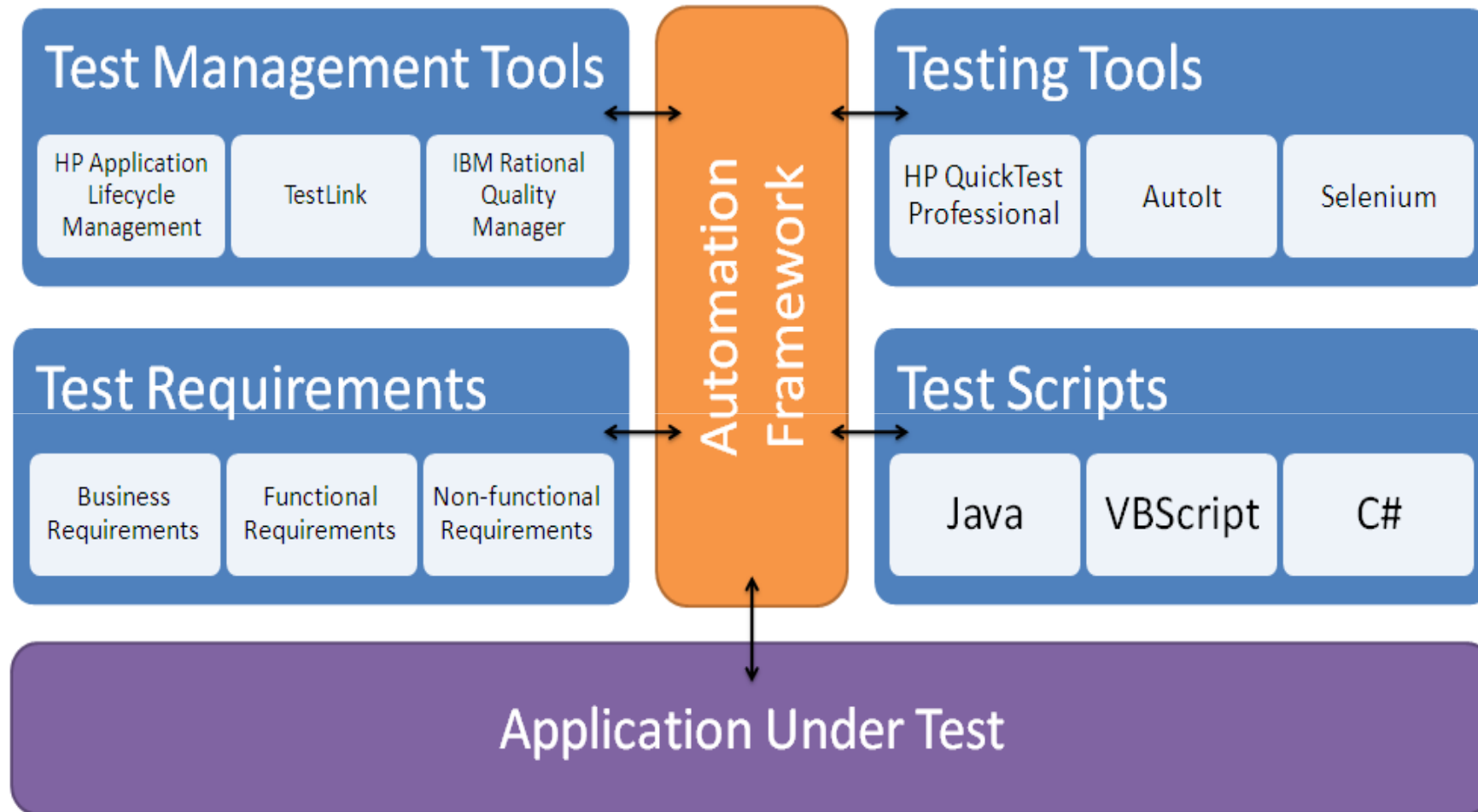
## Implementation:

### ★ Framework integrating:

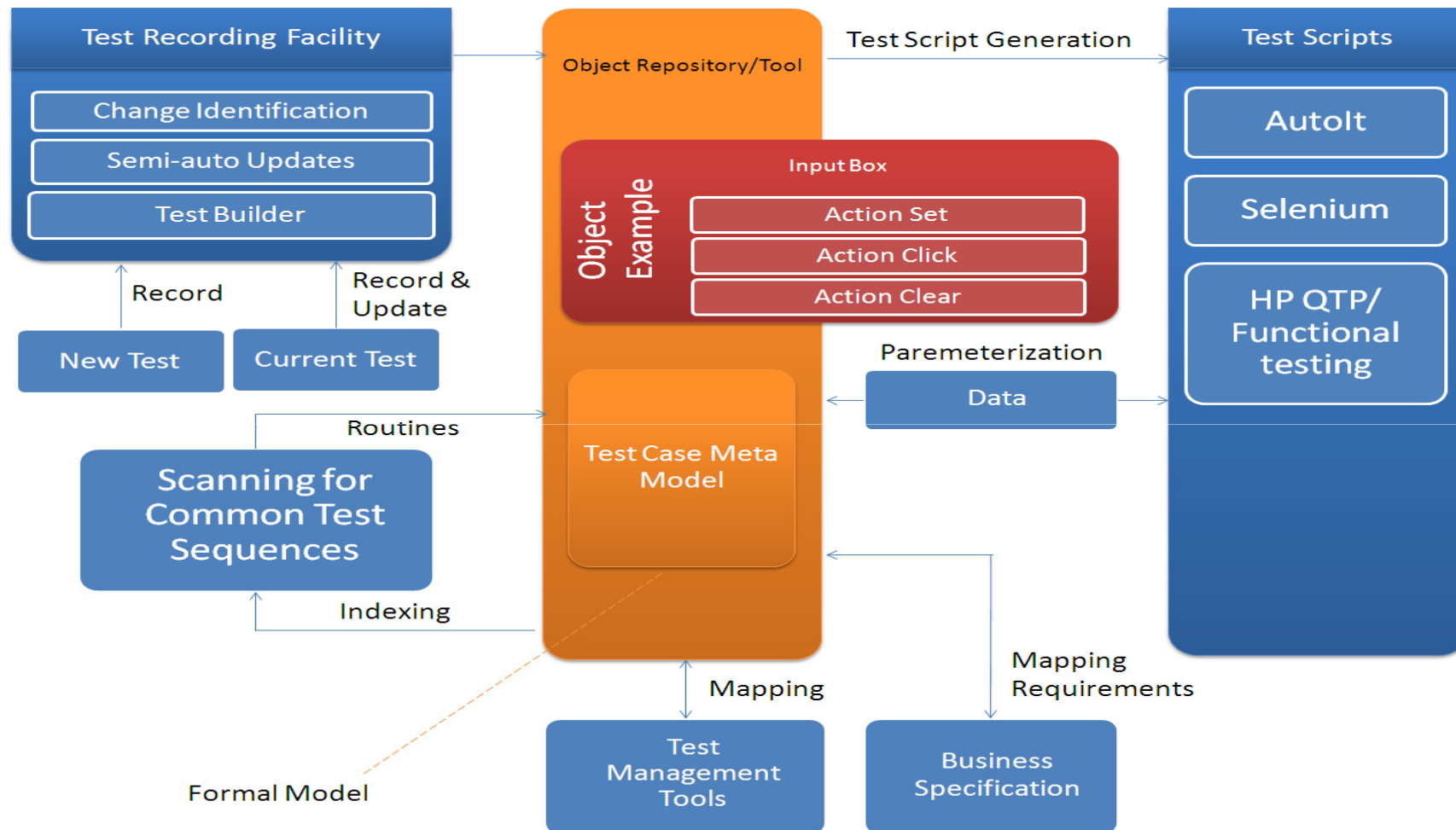
- Requirements
- Application under test
- Testing tool, Test management tool
- Test scripts



# Framework H-L architecture



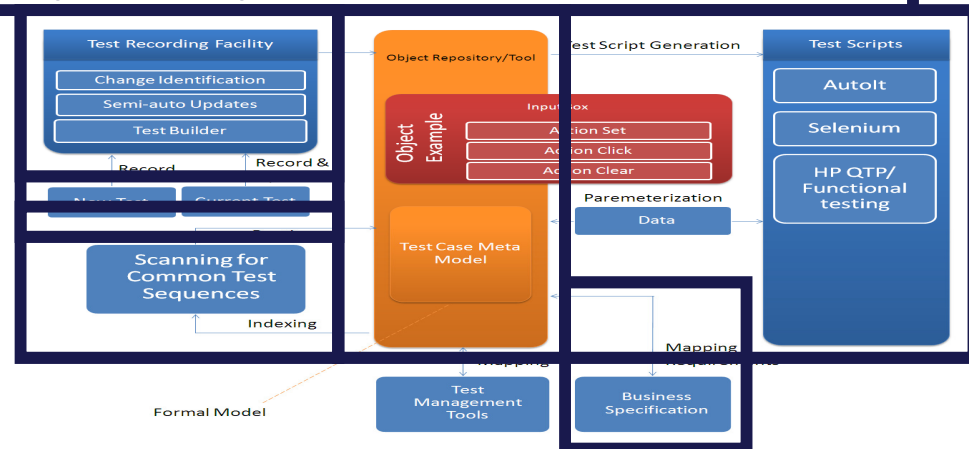
# The overall process



# Process in detail (1/2)

## Preparation of test scripts:

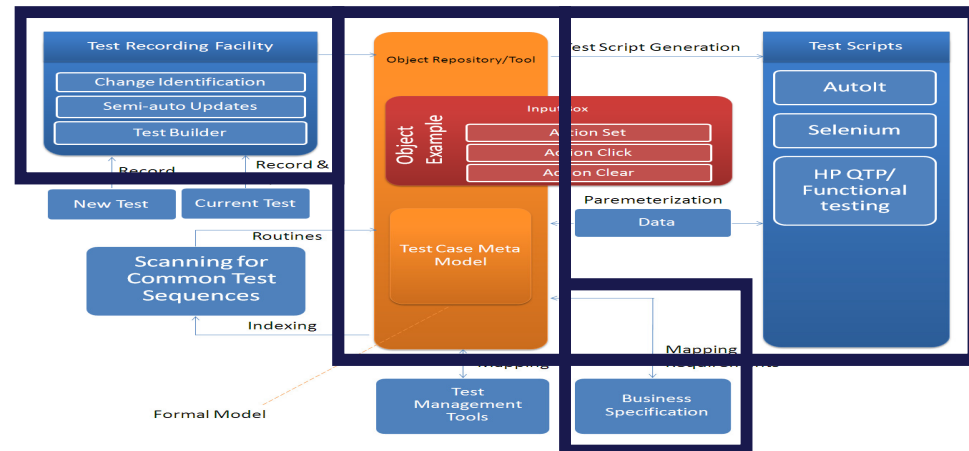
- Scan the application under test and store the GUI elements to the repository
- Map the requirements to the repository
- Record test script, store the script in meta-model
- Identify common (reusable) parts of the scripts
- Generate the physical test scripts for particular tool



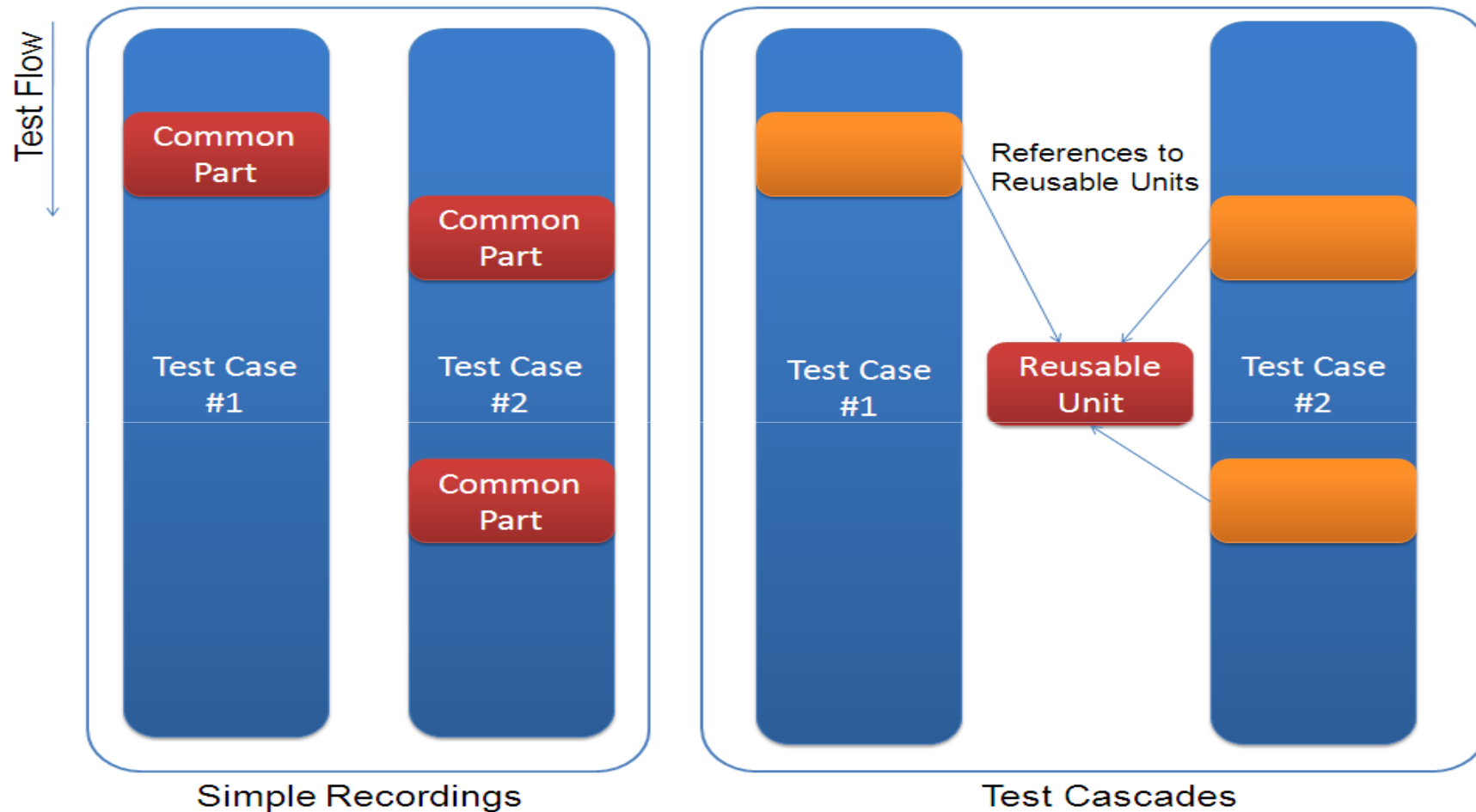
# Process in detail (2/2)

## Maintenance of test scripts:

- Requirements (specification) change:
  - Propagation to the model, identification of scripts to update
  - Generate updated physical test scripts
- Application under test changes:
  - Re-scan of GUI elements, identification of delta and scripts to update
  - Generate updated physical test scripts



# Identification of common parts



# Summary

---

## How easier maintenance is achieved?

- ★ Identification of script common parts
- ★ Automated re-generation of test scripts
  - When application under test changes
  - When requirements (specification) change
- ★ Generation of physical test scripts from meta-model repository for various test tools (→ platform independence)

## Framework flexibility:

- ★ Possibility of manual creation and maintenance of test scripts (in repository) as alternative to automated way
- ★ Implemented as platform-independent and open solution



**Conclusion**



# Conclusion

---

## Current outputs:

- ★ Principle of the solution
- ★ Architecture of the automation framework
- ★ Formal model being created

## Future work:

- ★ Detailed definition of the formal model
- ★ Experimental verification
  - Implementation of the framework
  - Application on real projects – comparison with standard test automation approaches



**Thank you for your attention**

filipma2@fel.cvut.cz  
buresm3@fel.cvut.cz