



Experience of generating Automated Tests for integration testing

Vjacheslav Lukashevich

TAPOST 2015

Who Am I



Vjacheslav Lukashevich

Tester

Test Lead

Developer

My Testing Interests

Agile testing

**Context
Driven
testing**

**Advanced
Test
Automation**

What I do

E2E testing

GUI testing

**Testing
strategies**

Test Tools

**Service/API
testing**

Test Automation

**Performance
testing**

Test Data

**Combinatorial
testing**

Education

■ Story

- Project and testing mission
- Testing and test automation challenge
- Solution – Automated Tests Generation
- Automation sub-project
- Project results
- Lessons I have learned

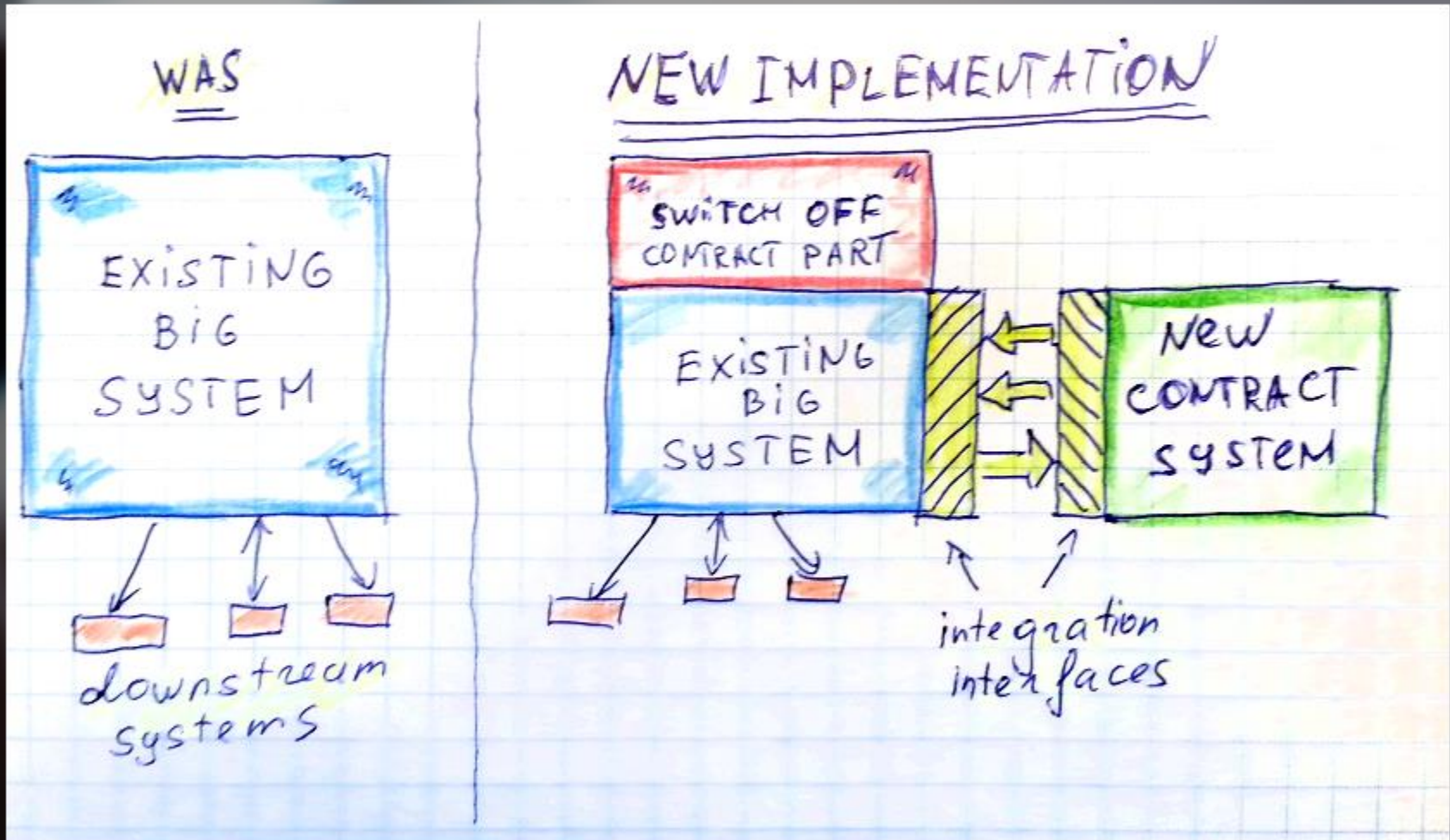
A nighttime aerial photograph of a city, likely Riga, Latvia. The image shows a wide river in the foreground, with a large cable-stayed bridge spanning it. The bridge has a tall, illuminated pylon and several stay cables. In the background, there are modern buildings, including one with a distinctive triangular facade. The foreground is filled with historic, multi-story buildings with red-tiled roofs and white walls. The overall scene is illuminated by city lights, creating a warm, golden glow.

PROJECT AND TESTING MISSION

Contract Management system project 1st Release

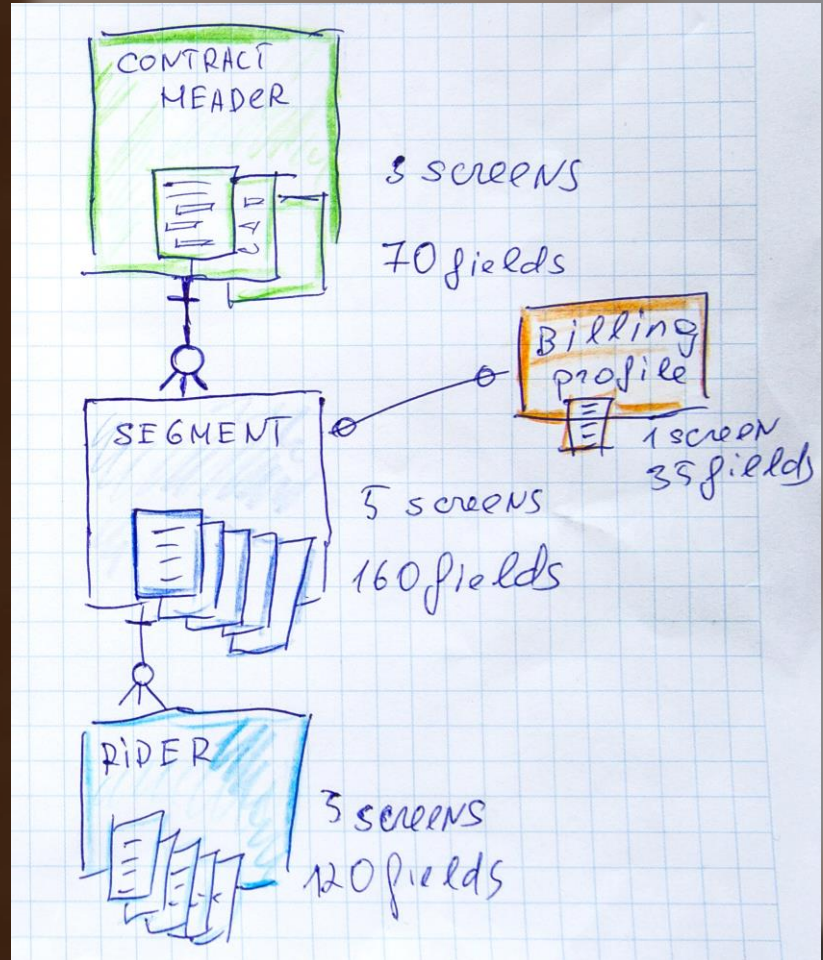
- Application details
 - Global Reinsurance business
 - Intranet Web application
 - Data capture and management
 - Replacing and extending part of existing system
- Project setup
 - Agile development, 2 weeks iterations
- Technology
 - Web frontend, Java Server Side
 - EJB integrations
- Test Automation tools
 - Framework for integrated testing (FIT)

IT landscape transformation project



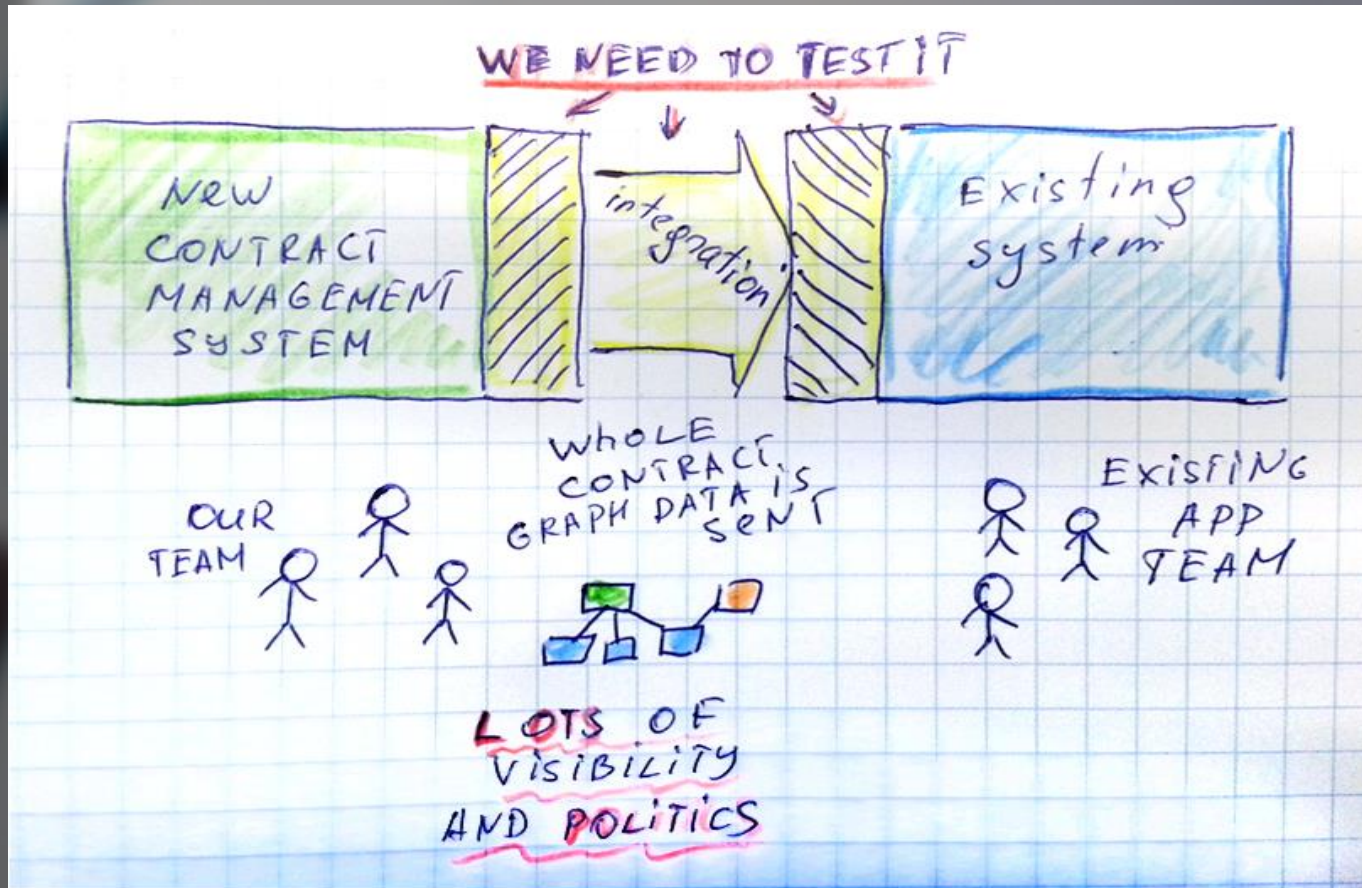
- The reinsurance contracts are
- BIG

- Each contract can have a lot of data
- There are a lots of data variations
 - >100 business rules for data

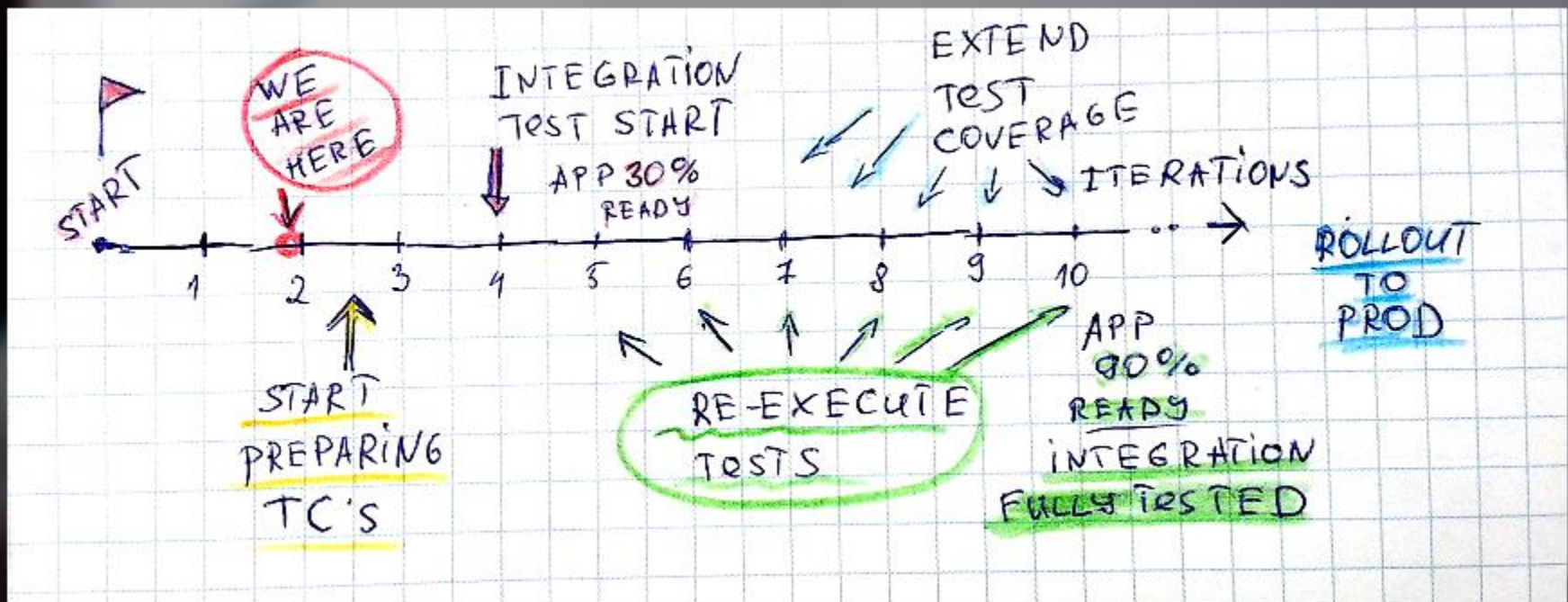


Mission – Test Integration between systems

Whole contract object is sent via integration interface



Integration testing plan

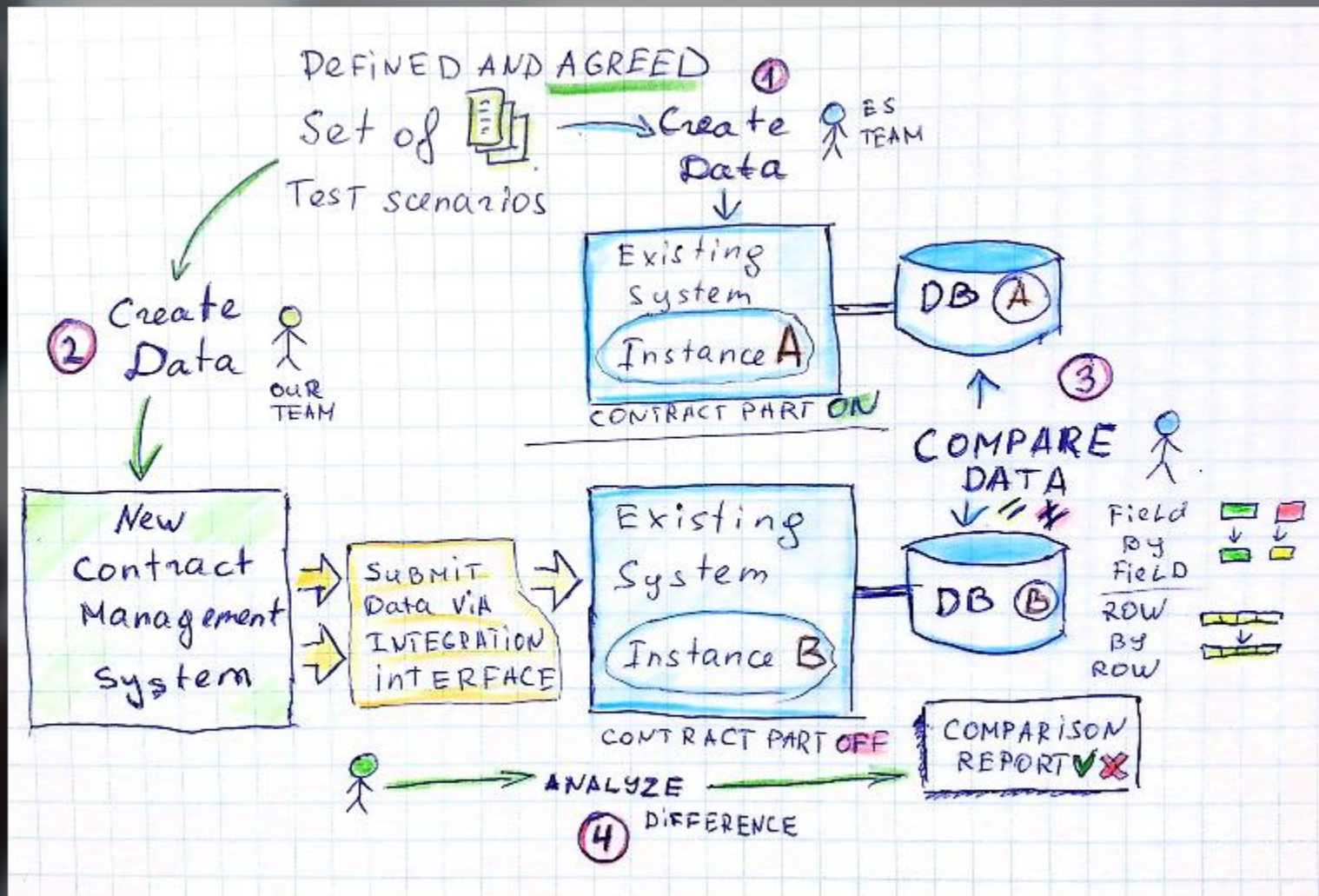




■ Lesson - Legacy system gives ■ lot of opportunities

- Different test oracles exist!
 - Working system
 - Documentation
 - Testing artifacts

Approach for integration testing



Contract Management regression test suite received from existing team

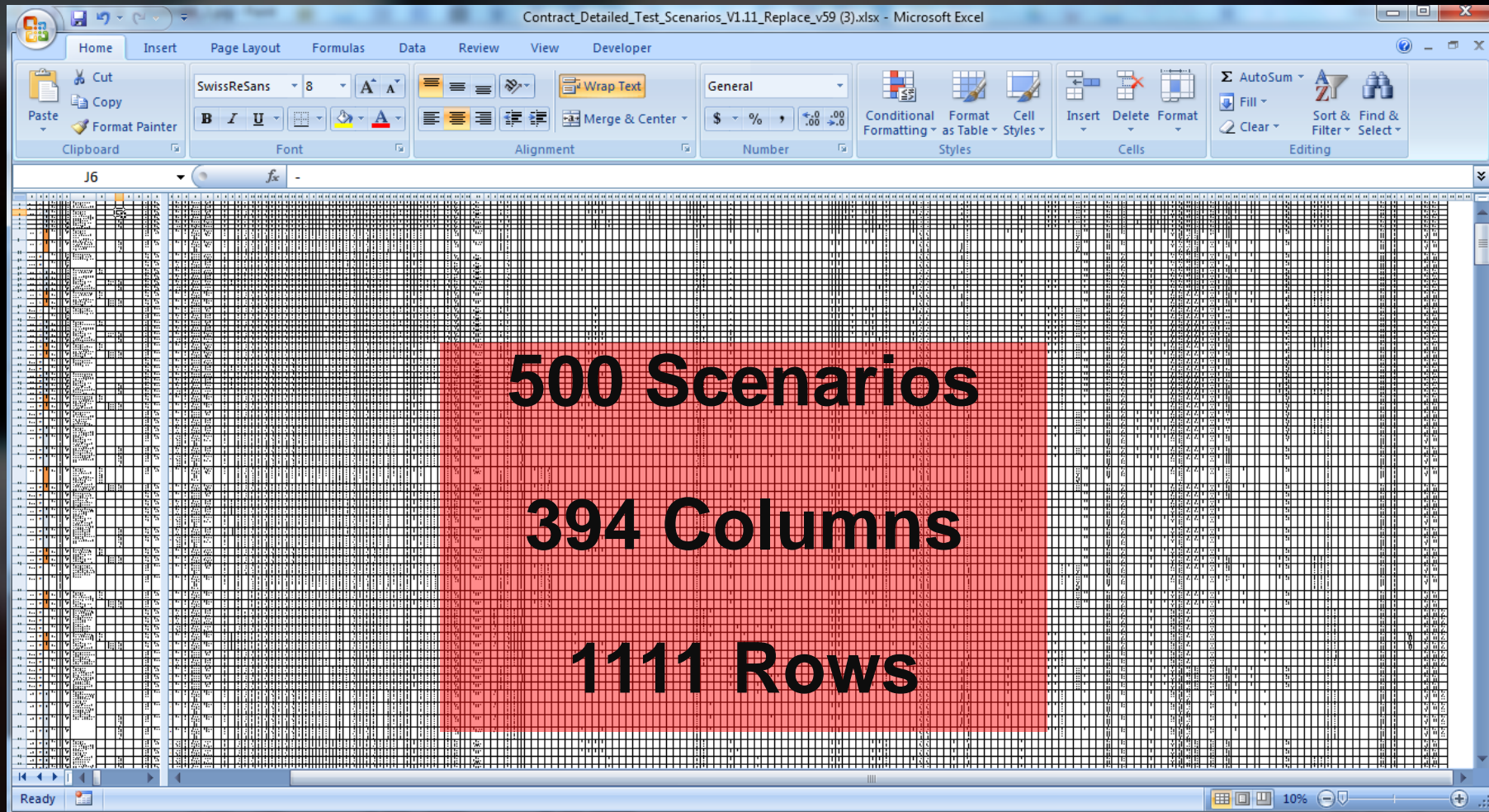
- Denormalized data, 1 scenario = one to many rows

	TC No	Priority	Test case Name	Amendment Type	Individual / Group	Treaty Cession Type	Business Unit	Segment Type	Life Health	Reinsurance Premium Type	Reinsurance Pr
2	1	Very High	Create a new individual Assmued treaty with 2 segments	-	Individual	Assumed	US	Life	Life	Coinsurance	Credit - Life
3	1	Very High		-	Individual	Assumed	US	Critical Illness	Health	Yearly Renewable Term	Credit - Critical I
4	2	High	Create a Treaty/Segment Change amendment by adding 2 segments for the newly created individual Assmued treaty	Treaty/Segment Change	Individual	Assumed	US	Life	Life	Coinsurance	Catastrophe -
5	2	High		Treaty/Segment Change	Individual	Assumed	US	Income protection	Health	Yearly Renewable Term	Waiver of Pren
6	3	High	Set Up: Individual Assumed Treaty with one segment. Process: Do a Treaty/Segment Change amendment by adding a new segments with max. treaty effective date	Treaty/Segment Change	Individual	Assumed	US	Life	Life	Yearly Renewable Term	Catastrophe -
7	4	Very High	Create a new Group Assmued treaty with 2 segments	-	Group	Assumed	US	Life	Life	Coinsurance	Interest Sensitiv
8	4	Very High		-	Group	Assumed	US	Critical Illness	Health	Yearly Renewable Term	Credit - Critical I
9	5	High	Create a Treaty/Segment Change amendment by adding 2 segments for the newly created Group Assmued treaty	Treaty/Segment Change	Group	Assumed	US	Critical Illness	Health	Yearly Renewable Term	Critical Illness - Sta
10	5	High		Treaty/Segment Change	Group	Assumed	US	Income protection	Health	Yearly Renewable Term	Income Protection - I
11	5	High									



PROBLEMS AND CHALLENGES

Scenarios have LOTS of data..

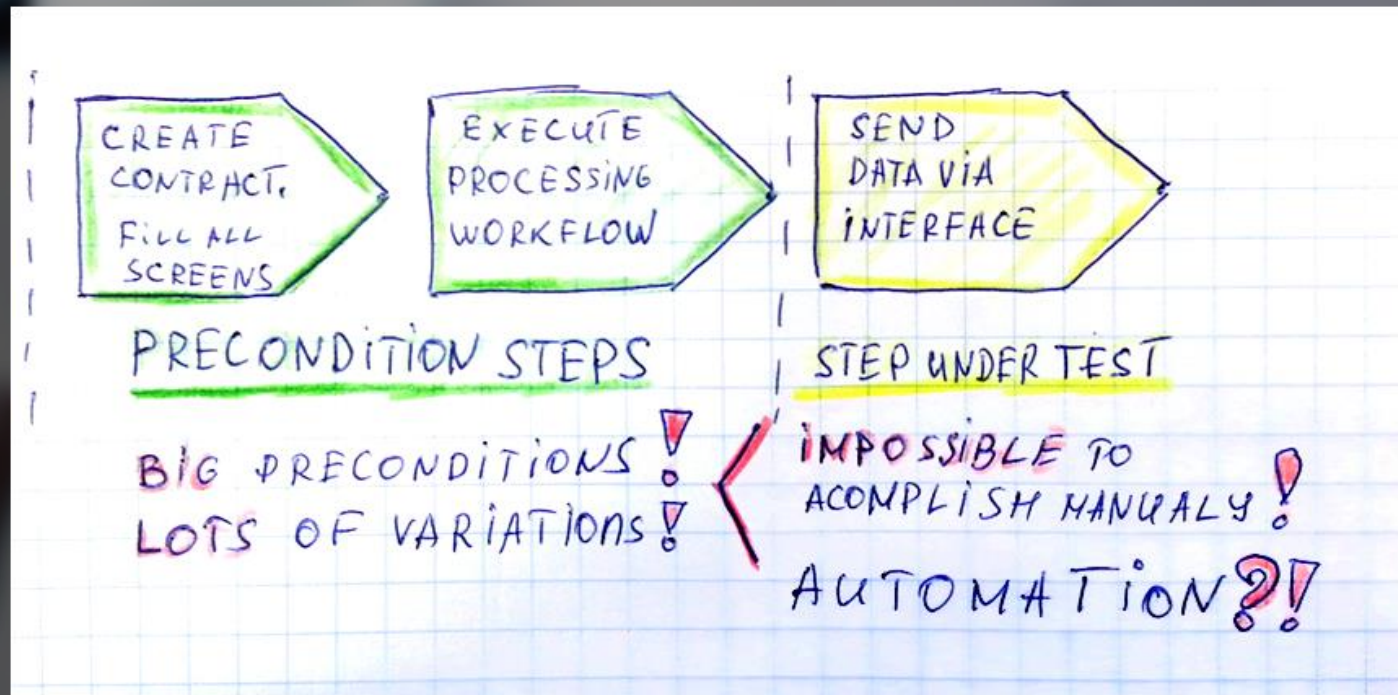


Problem#1

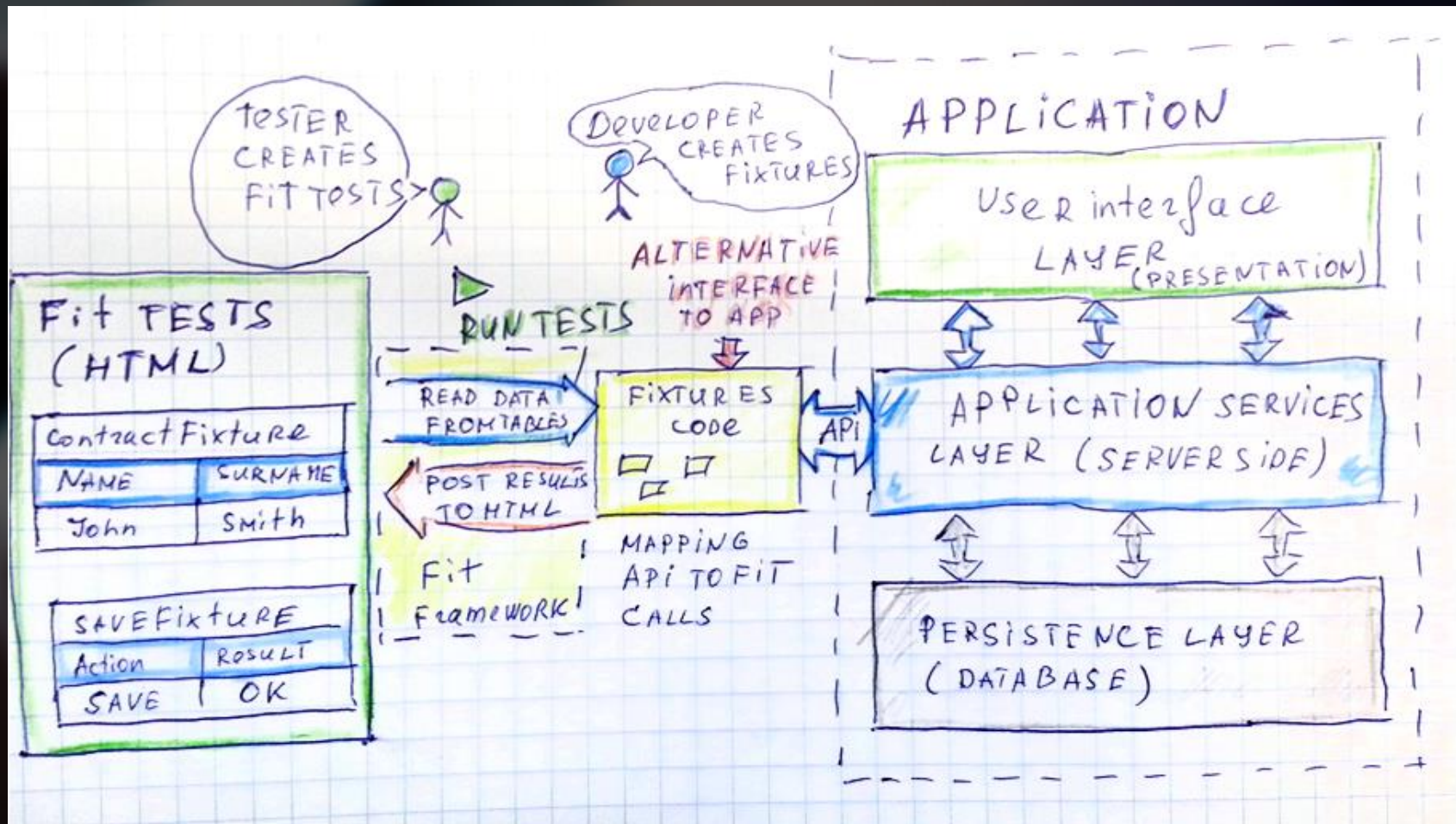
- Each scenario is BIG

Some are VERY BIG....

We need to execute lots of steps to create and fill contract screens



A little bit about FIT tool



Framework for Integrated Test (FIT)

Design time

.../PersonTableEnterFixture		
Name	Surname	DateOfBirth
Albert	Einstein	14.03.1879
Isaac	Newton	25.12.1642
Cristopher	Columbus	14.09.1451

.../PersonTableActionFixture		
Save()		
OK		

.../PersonTableViewFixture		
Name	Surname	DateOfBirth
Albert	Einstein	14.03.1879
Isaac	Newton	25.12.1642
Cristopher	Columbus	14.09.1451

Execution results

.../PersonTableEnterFixture		
Name	Surname	DateOfBirth
Albert	Einstein	14.03.1879
Isaac	Newton	25.12.1642
Cristopher	Columbus	14.09.1451

.../PersonTableActionFixture		
Save()		
OK		

.../PersonTableViewFixture		
Name	Surname	DateOfBirth
Albert	Einstein	14.03.1879
Isaac	Newton	<i>expected</i> 25.12.1642
		<i>actual</i> 25.12.1641
<i>missing</i>	Columbus	14.09.1451
Cristopher		
<i>surplus</i>	Vespucci	09.03.1454
Amerigo		

■ FIT Scripts are business scenarios

- No programming logic inside script (IF, loops)
 - Top down execution
- Values are mostly hardcoded inside FIT script
 - Reusable parts can be parametrised

■ FIT scripts development

- Test engineer compose FIT script using any suitable HTML editor

Problem#2

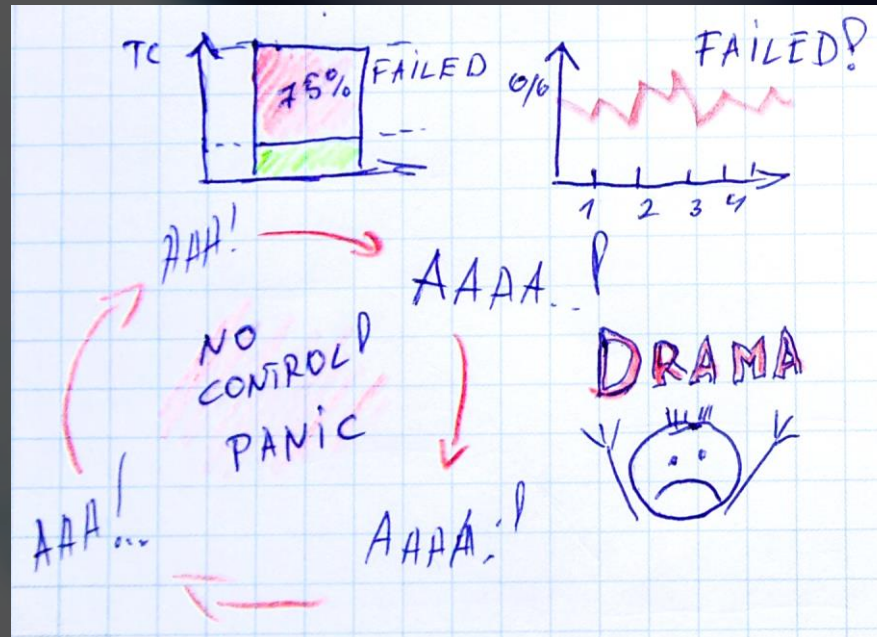
Scripts volume is too BIG

- Too many
 - Scenarios
 - Steps needed
 - 300-1500 steps per scenario
 - Data variations
 - Parametrisation required

**Too much manual effort
and overcomplexity
for test automation**

Problem#3

- No control over automation
 - We can't support this amount of tests
 - Tests FAIL constantly and there is no confidence in our testing



How test automation status looked like on our daily standup

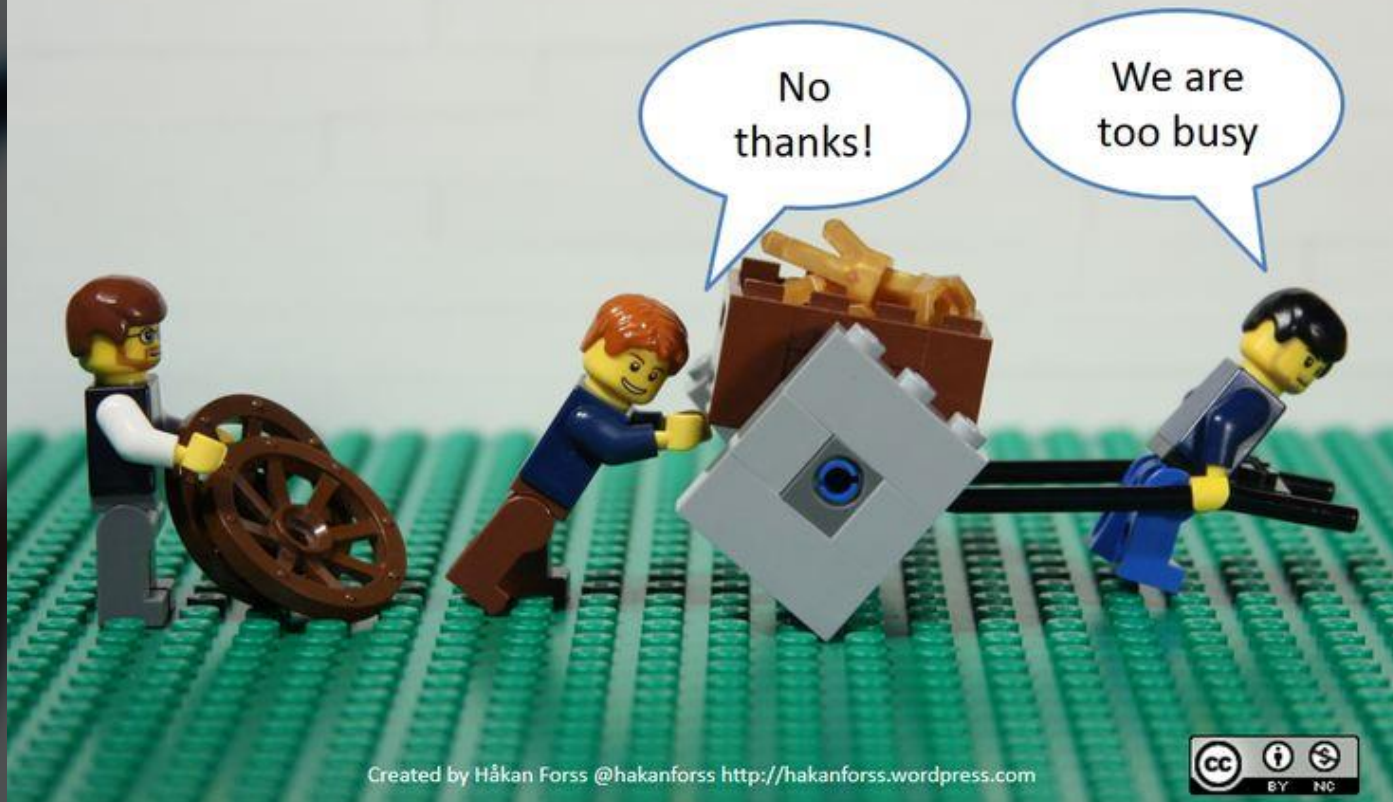


A nighttime aerial photograph of a city, likely Riga, Latvia. The image shows a wide river in the center, with a prominent cable-stayed bridge spanning it. The bridge has a tall, white, A-frame pylon. To the left, there are modern buildings, including one with a distinctive, illuminated, angular facade. To the right, the city's historic center is visible, featuring numerous multi-story buildings with red-tiled roofs and a large, ornate church with a tall, dark spire. The word "SOLUTION" is overlaid in large, white, sans-serif capital letters across the middle of the image.

SOLUTION

- It could be difficult to step out
- when you are deeply in process

Are you too busy to improve?



■ Maze from different viewpoints

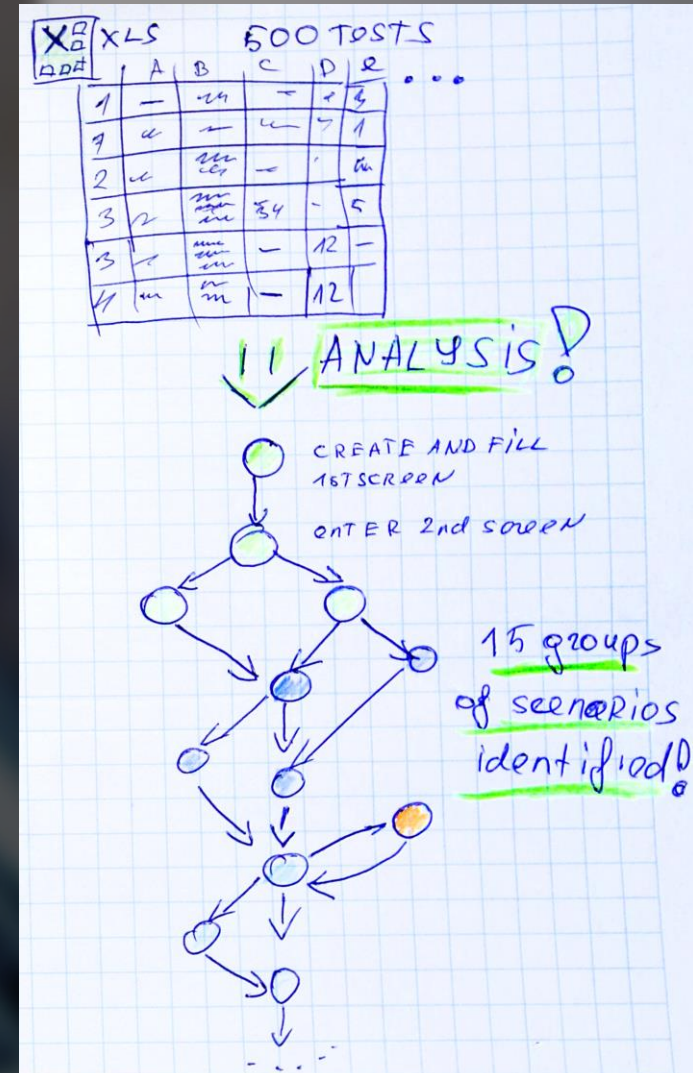


VS



■ Analysis of test data

- 1st part of all scenarios is very similar!
- Many scenarios have groups of similar logic!



■ FITs are HTML!

- HTML generation from templates is common approach in industry



- Let's automate the test automation creation!



- Team assigned to prototype
- FIT generation

Developer



Tester



Template engine – FMPP

- FMPP - FreeMarker-based text file PreProcessor
- General-purpose text file preprocessor tool.
- Java
- Open source



■ Test Data related tasks

- Analyze scenarios and define test generation rules
- Align test data for generation
 - Align column names
 - Align field values
 - Add additional fields to support generation logic
 - Scenario group ID
 - Object ID in the contract graph
 - Action flags

Generation rules

#	IF	THEN
1	First row of a scenario	Fill Treaty Header, Additional details, Non-Contractual details screens with field values from this row
2	Field value = "-"	Consider this as empty value
3	Last row of the scenario	Re-save Claims and Segments for all segments/riders ever used in the scenario as " <i>Completed</i> " or " <i>Draft</i> " based on the values in the ClaimsCompleted for Claims or AdminCompleted for Admin
4	BenefitType has list of values separated by comma (,)	Consider this as a multiple Benefit values for the Additional Benefit multi value list
..

Aligned data

TCNo	Groups	ProfitCenter	TestcaseName	TreatyEffectiveD	AmendmentEf	Individual_Group	TreatyCessionT	BusinessUr	WorkflowVi	Segment	SegmentType	LifeHealth	ReinsurancePremium
TC No	Group s	Profit Center	Test case Name	Treaty Effective Date	Amendment Effective	Individual / Group	Treaty Cession Type	Business Unit	Workflow View	Segment ID	Segment Type	Life Health	Reinsurance Premium Type
1	0	US Life & Health PCR26	Create a new individual Assumed treaty with 2 segments	1/1/2010	-	Individual	Assumed	United States	US Process	S1	Life	Life	Coinurance / Original Term / Net Level Premium
1	0	US Life & Health PCR26		1/1/2010	-	Individual	Assumed	United States	US Process	S2	Critical Illness	Health	Yearly Renewable Term / Risk Premium
2	2	US Life & Health PCR26	Create an individual assumed treaty with two segments. (S1 & S2) Create a "Treaty/Segment Change" amendment by adding another 2 new segments(S3 and S4)	1/1/2010	-	Individual	Assumed	United States	US Process	S1	Life	Life	Coinurance / Original Term / Net Level Premium
2	2	US Life & Health PCR26		1/1/2010	-	Individual	Assumed	United States	US Process	S2	Critical Illness	Health	Yearly Renewable Term / Risk Premium
2	2	US Life & Health PCR26		-	3/3/2010	Individual	Assumed	United States	US Process	S3	Life	Life	Coinurance / Original Term / Net Level Premium
2	2	US Life & Health PCR26		-	3/3/2010	Individual	Assumed	United States	US Process	S4	Income Protection	Health	Modified Coinurance
3	1	US Life & Health PCR26	Create an individual assumed treaty with one segment(S1).Do a Treaty/Segment amendment with addition of a new segment (S2)	1/1/2008	-	Individual	Assumed	United States	US Process	S1	Life	Life	Yearly Renewable Term / Risk Premium
3	1	US Life & Health PCR26		-	4/10/2008	Individual	Assumed	United States	US Process	S2	Life	Life	Yearly Renewable Term / Risk Premium
4	0	US Life & Health PCR26	Create a new Group Assumed treaty with 2 segments	2/1/2010	-	Group	Assumed	United States	US Process	S1	Life	Life	Coinurance / Original Term / Net Level Premium

■ Generator implementation tasks

- Create templates for screens
- Define composition/chains of templates for scenario group
- Implement test generation rules

■ FMPP example

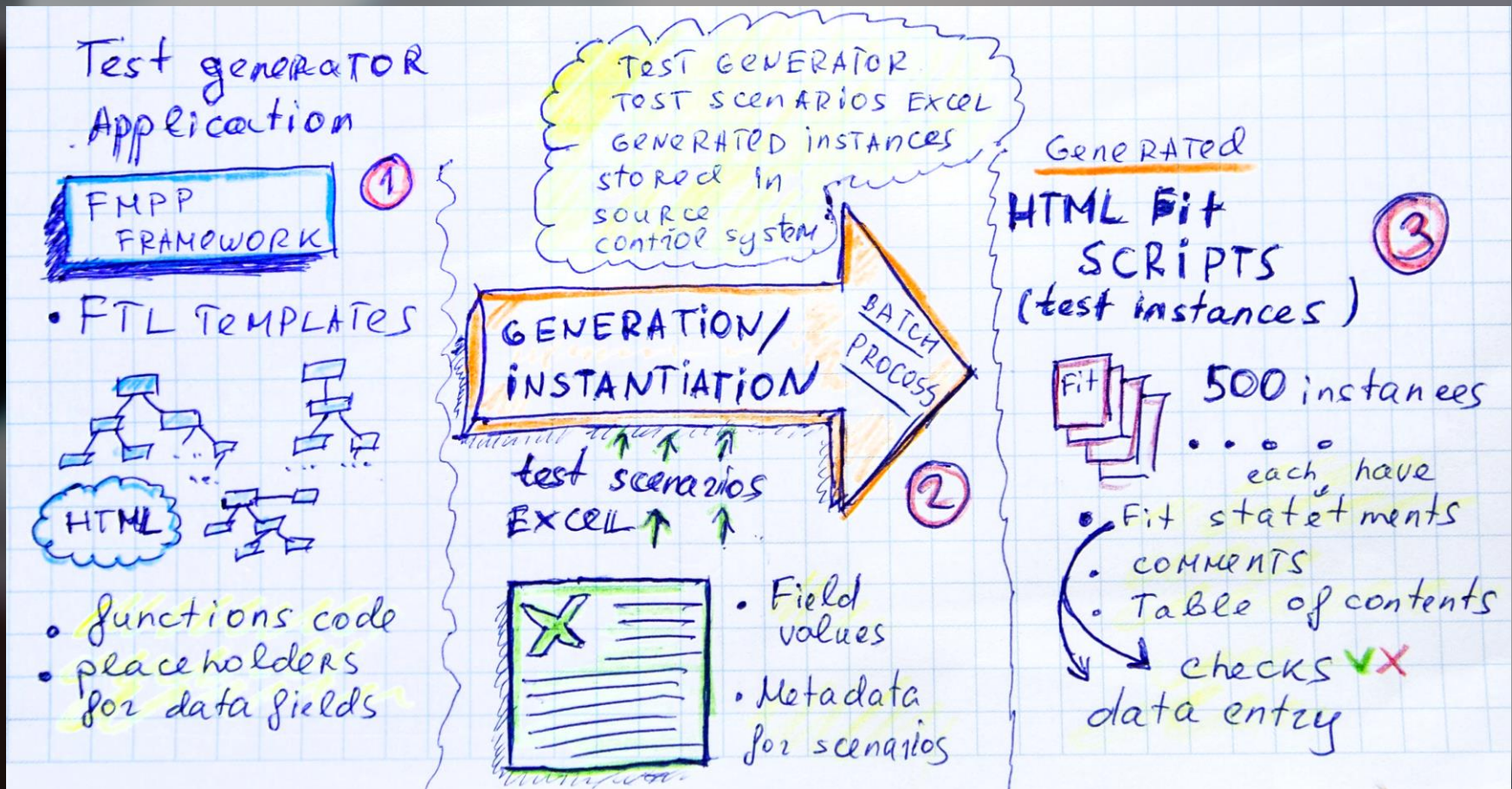
■ createNewTreatyHeader.ftl

```
<table border="1">
  <tr>
    <td colspan="3">com.ctco.tmr.fit.DataEnterFixture(TreatyDetailsFitContext)</td>
  </tr>
  <tr bgColor="lightblue">
    <td>contractName</td>
    <td>contractNameVersionEffectiveDate</td>
    <td>contractNameCessionType</td>
  </tr>
  <tr>
    <td><span class="fitVar">${var_contractNameeyName}</span></td>
    <td><span class="fitVar">${var_contractNameEffective}</span></td>
    <td><span class="fitVar"><#if var_Empty == fit.contractNameCessionType>@@NULL@@</#if></span></td>
  </tr>
</table>
```

■ template_group_11.ftl

```
<#-- This is New Treaty -->
<#if var_IsHeader>
  <#include "../common/createNewTreaty_BaseInformation.ftl">
  <#include "../common/createNewTreaty_Login.ftl">
  <#include "../common/userPreferences.ftl">
  <#include "../common/createNewTreaty.ftl">
  <#include "../common/billingProfile.ftl">
</#if>
```


Test Generation



Test automation approach

HTML TEMPLATES

+

EXCEL TEST DATA

=

MODEL BASED TESTING

+

DATA DRIVEN TESTING

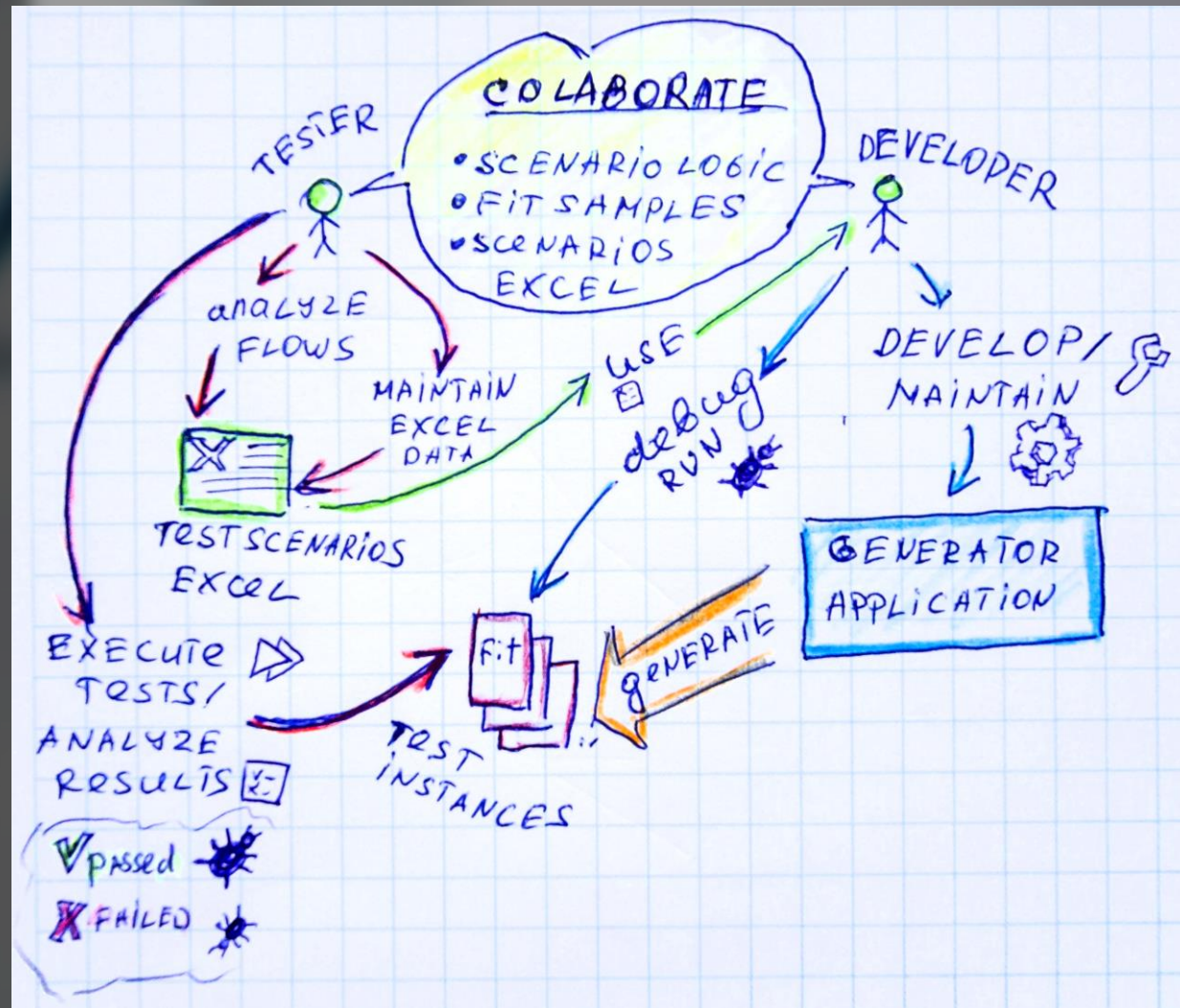
A nighttime photograph of a train station in Riga. In the background, a tall clock tower with the word "RĪGA" at the top is illuminated. A church spire is visible to the left. In the foreground, two passenger trains are stopped at the platform, their headlights on. The scene is lit by streetlights and the train's lights.

AUTOMATION SUB-PROJECT

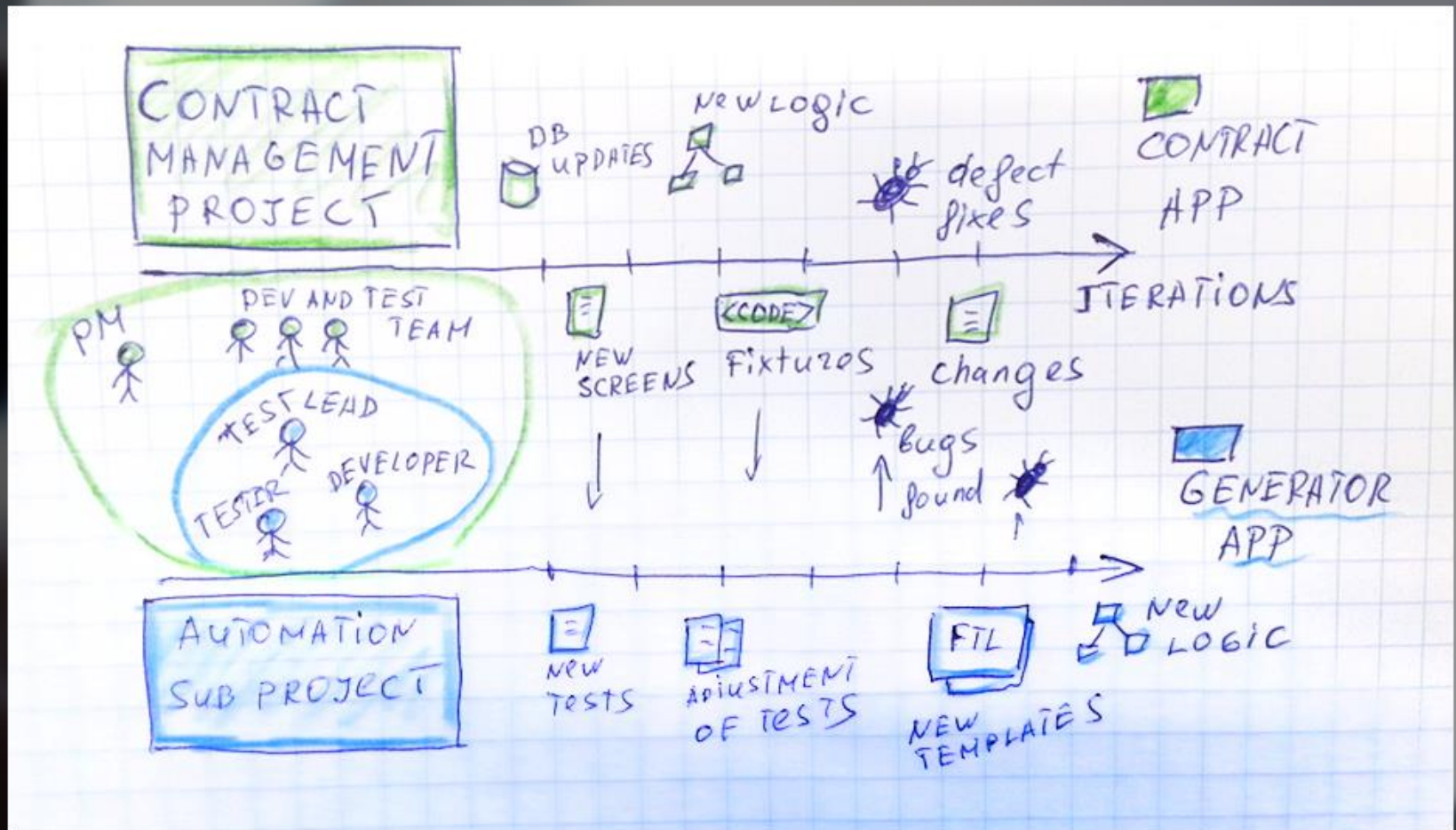
Automation sub project

- Dedicated team
 - Test lead
 - Tester
 - Developer
- Development process
 - Issue tracking
 - Scope management
- Testing process
 - Execution on local and integrated environment

Tester and developer role



Automation Sub Project



■ Executing of tests

- Local environment
 - Debugging of automated tests
 - Runs fast
- Integrated environment
 - Goal to find problems with integration
 - After problem fixes we were re-executing failed scenarios

■ Findings

- Defects in application
- Errors on the interface side
- Errors in the automation code
- Errors in the specifications
- Errors in legacy application
- Functionality Updates in CM
(Maintenance)



PROJECT RESULTS

■ Automation is manageable

- Scope known
- Implementation controlled
- Maintenance controlled
- Tests are up to date to specific version
- Constant and intensive feedback from testing

■ Result for 1st Release

- 100 small templates VS 500 big individual tests
- Automated all before the plan
- Resources spent are in line with budget
- Acceptance testing done automatically
- Caught lots of problems
- Regression test suite is ready

■ Results after several releases

- Tests are still working
- Maintenance done with reasonable resources
- Suite was slightly extended with new tests
- Approach was adopted in other projects

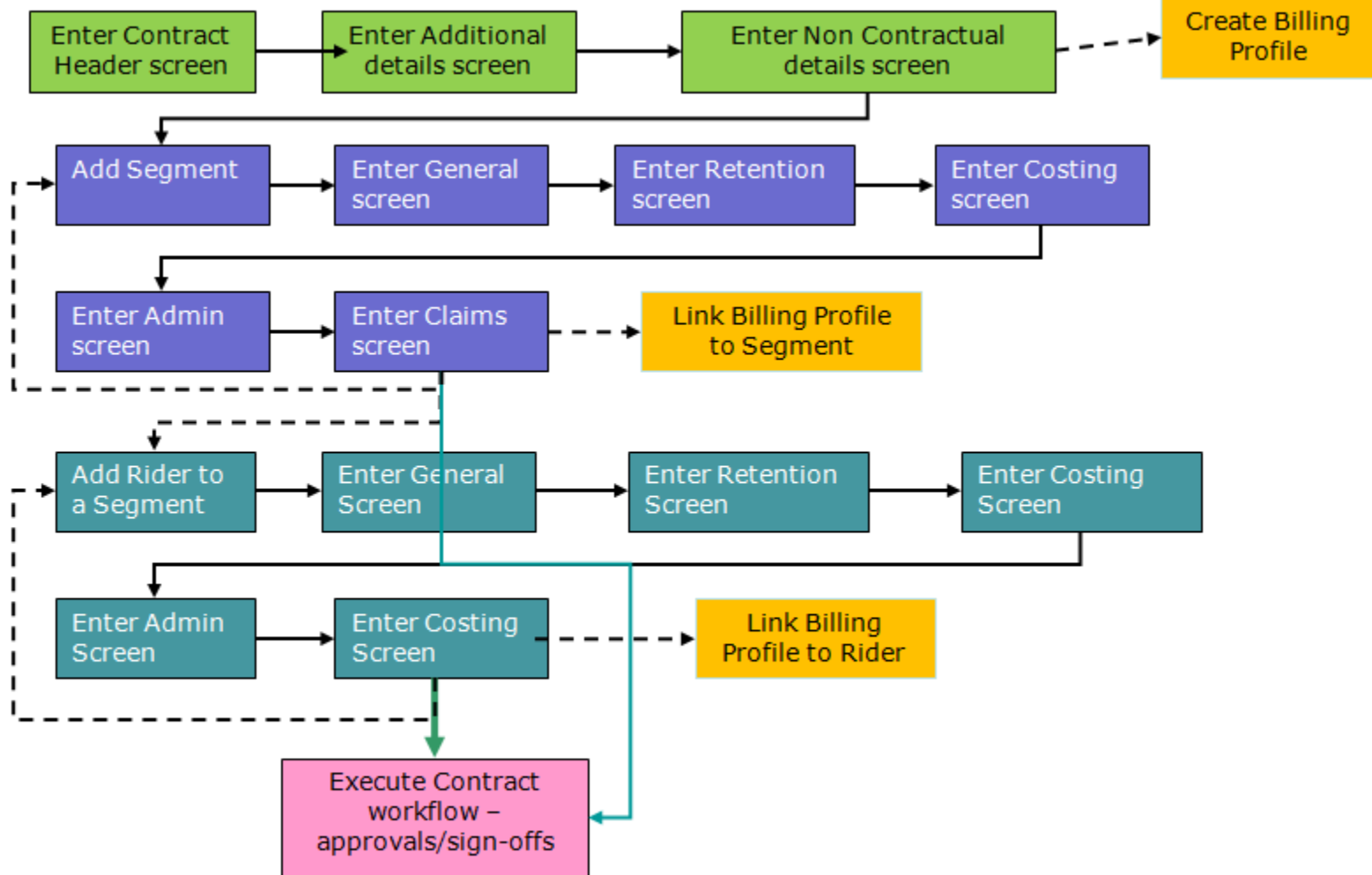
■ WHAT I HAVE ■ LEARNED



■ Lesson #1

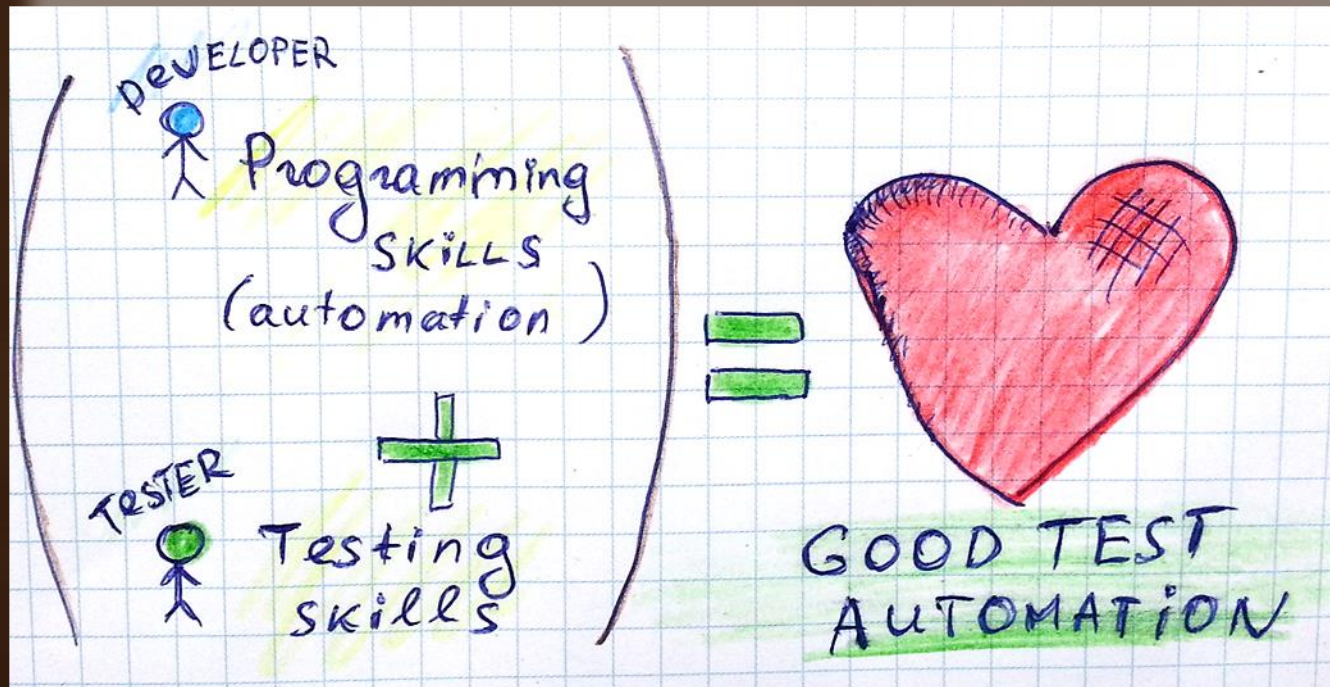
- Test Generation is VERY effective for applications that can be represented as a workflow graph

Scenario graph sample from our project



Lesson #2

- Cooperation of developer and tester for Test Automation implementation is VERY productive!



■ Lesson #3

- Managing automation as sub-project is VERY GOOD model for dealing with complex automation tasks
 - People assigned
 - Planning, scope, progress managed

Lesson #4

- Managing test data in excel format is VERY convenient!
 - Find tests that use the specific data for problem analysis
 - Easy maintenance of test data
 - Add new columns
 - Bulk updates

	K	L	M	N	O	P
	WorkflowView	Segment	SegmentType	LifeHealth	ReinsurancePremium	ReinsuranceProduct
	Workflow View	Segment/Rider	Segment Type	Life Health	Reinsurance Premium Type	Reinsurance Product
	US Process	S2	Life	Life		
	US Process	S1	Life	Life		
	US Process	S2	Critical Illness	Health		
	US Process	S1	Life	Life		

Sort A to Z
Sort Z to A
Sort by Color

Clear Filter From "Reinsurance Product"
Filter by Color

Text Filters

- ☐ Income Protection - short term
- ☐ Interest Sensitive Life
- ☒ Office Overhead
- ☐ Term
- ☐ Total Disability Monthly Income
- ☒ Universal Life - Fixed
- ☐ Universal Life - Variable
- ☐ Waiver of Mortgage Interest
- ☐ Waiver of Pension Contribution

OK Cancel

■ Lesson #5

- Keep it simple and documented!
 - Keep business logic on the test data
 - Keep test data in simple form
 - Keep test generation logic as simple as possible
 - Document the solution and approach

■ Lesson #6

- Service/API based automation allows to execute
VERY BIG volume of testing

■ ...AND MANY OTHER ■ LESSONS..

Sometimes later...



Use
ALL OPPORTUNITIES
to make your testing better
and
GOOD LUCK
with your
TEST AUTOMATION

Questions?