The Road to Hell and Back

Patterns in Test Automation project failure & Recovery

Alon Linetzki, ISRAEL

The Best are **Getting Better**

Alon Linetzki www.Best-Testing.com alonl@sigist.org.il +972 (0)54 4848533

We are not the ASBEST conference…

Using the material inside this presentation

- You may use some of the material from this presentation under the following conditions:
  - Copyrights still belong to the authors described below, and you get a permission to use it only
  - When using material or part of the material from this presentation, you are obligated to reference the authors in this way:

"© Copyrights of Alon Linetzki, Best-Testing & Michael Stahl, Intel"

Alon Linetzki

- A Software Engineer, working both as a Testing and QA professional coach and as a business Integration/enhancement promoter,
- **Supporting and Enabling organizations to improve their product development and testing operation, by taking informed decisions, using product quality and process related information**
- Working in multi-national, multi-cultural, multidisciplinary operations and environments,
- Supporting engineers, team leaders, managers and executives,
- Global services are supported around the world,
- ISTQB® Foundation Agile Tester Extension co-Author

- ISTQB Agile Tester
- ISTQB Advanced TA
- ISTQB Advanced TM
- Scrum Master
- LQA ISO-9000
- TMMi
- CMMi
- TPI™
- System Analyst
- CMAP®
- … ☺

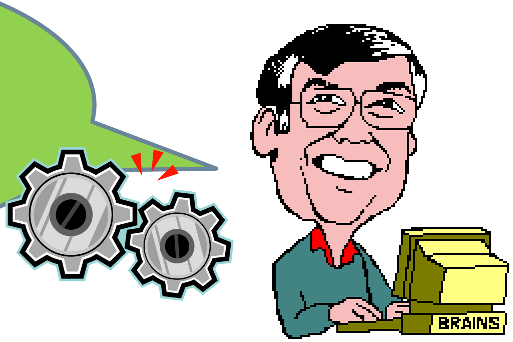
On the Menu…

- □ A common test automation project story
- □ Alert Signals
- □ Counter Measures
- □ Summary

Common test automation story

- Once upon a time…

I can automate
my work!...



Mr. Otto Mate

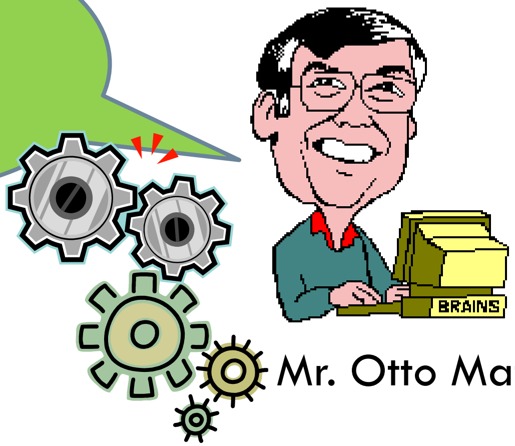
Cool!
Can you make it…

Otto’s Team

Show to friends…

Common test automation story

Piece of cake!

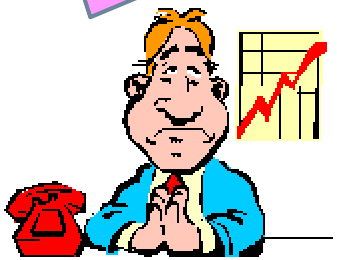


Mr. Otto Mate

Perfect!
Can you add…

Otto’s Team

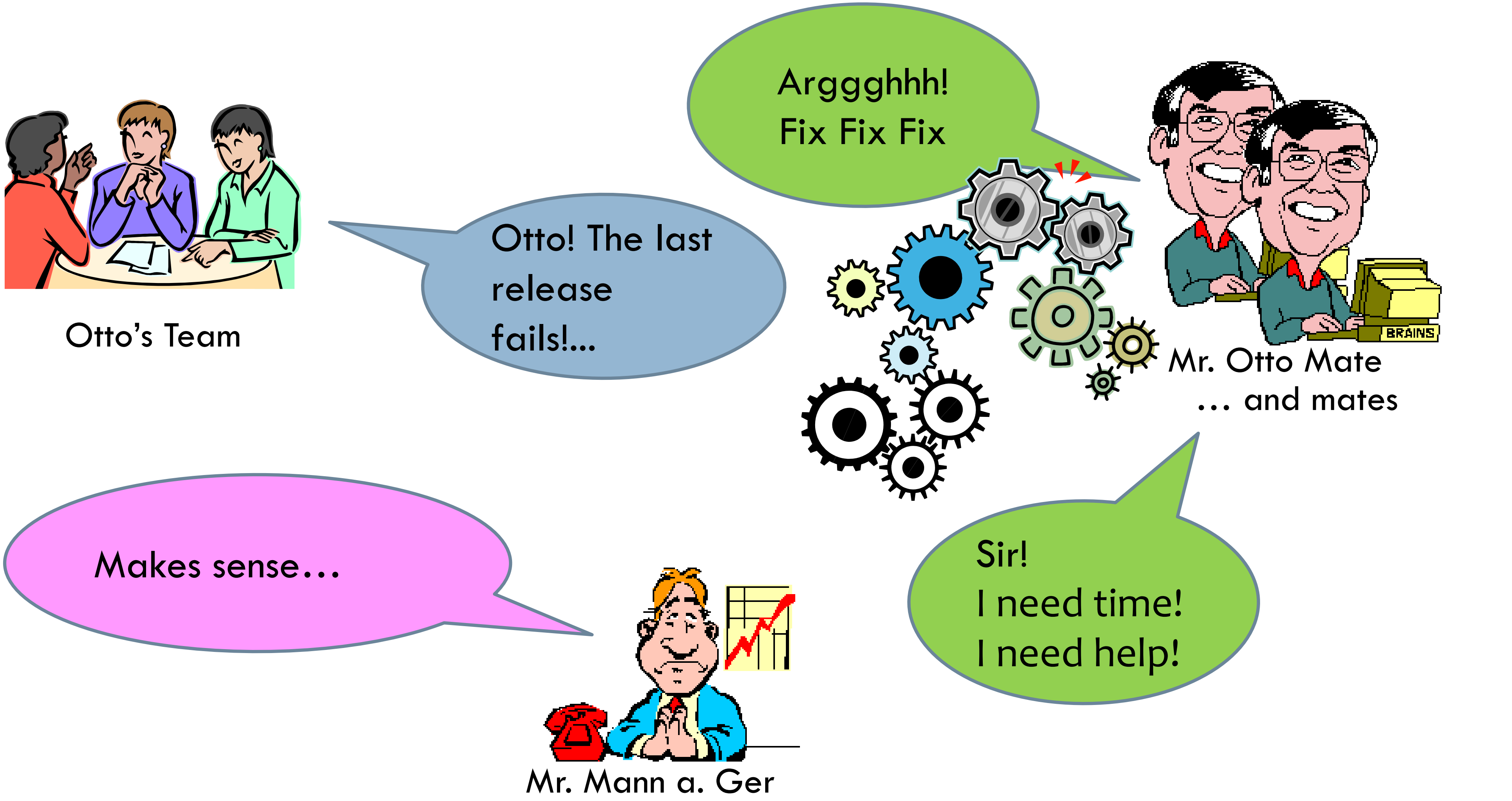
What a wonderful world!...



Mr. Mann a. Ger

Show to friends…

Common test automation story



Otto's Team

Otto! The last release fails!...

Argggghhh! Fix Fix Fix

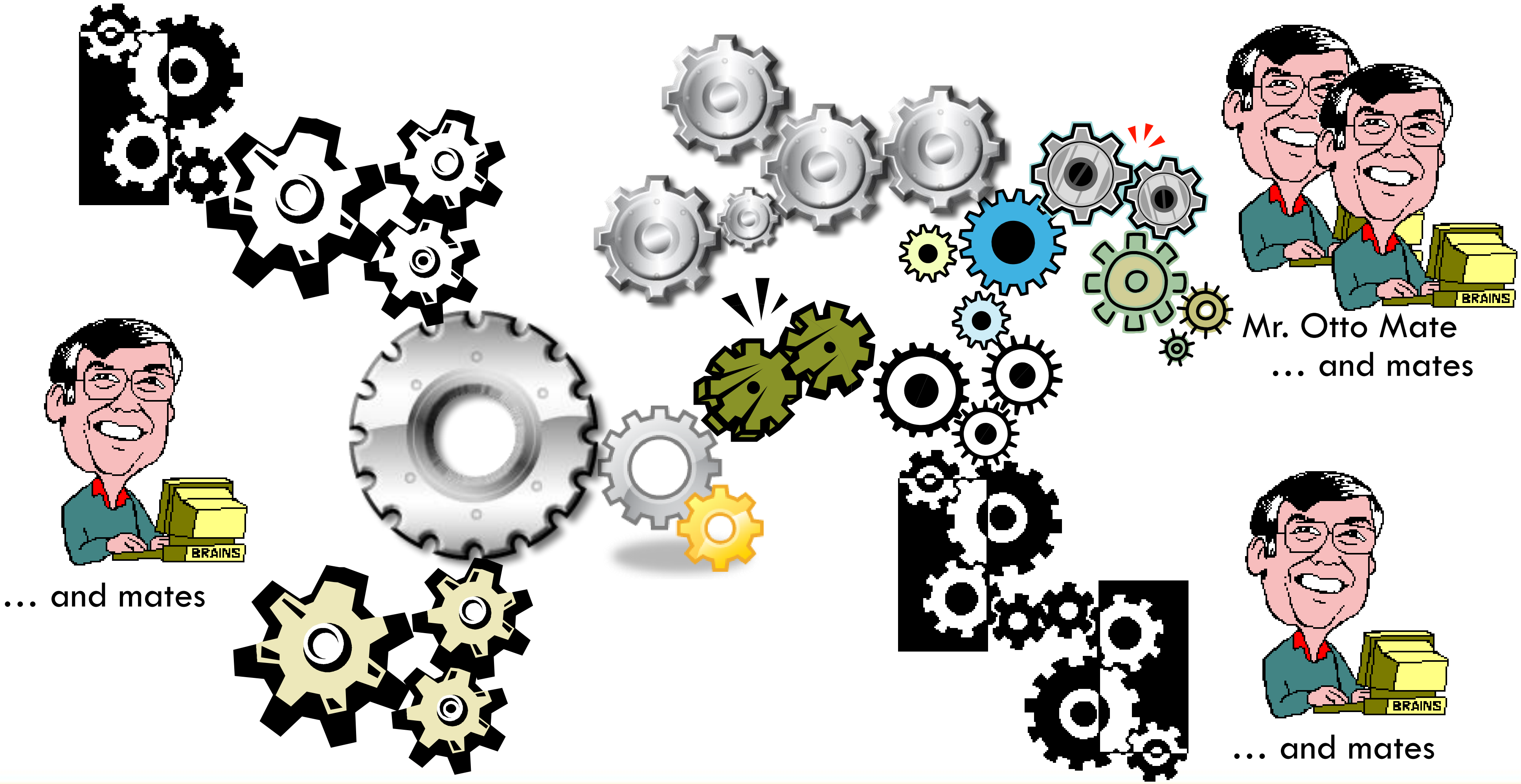
Mr. Otto Mate … and mates

Sir!
I need time!
I need help!

Makes sense…

Mr. Mann a. Ger

Common test automation story

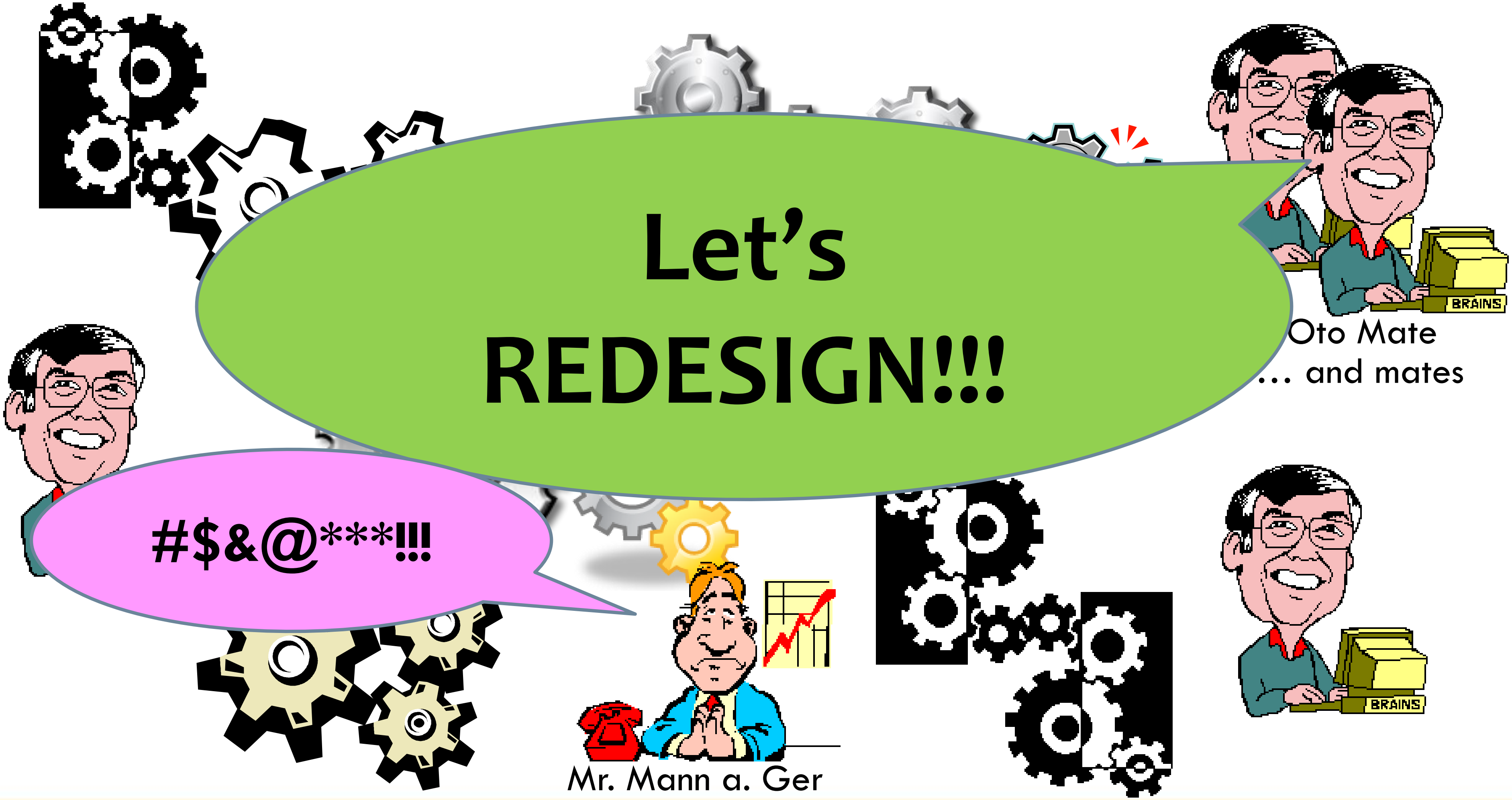


Mr. Otto Mate
… and mates

… and mates

… and mates

Common test automation story



Motivation

- So many times…
  - We let a sma[illegible]ion project [illegible]ushroom into a [illegible]e **framework…**
  - We allowed a [illegible]n adv[illegible]**ncontrolled**, as a mean to retai[illegible]ester…
  - We were suck[illegible] cy[illegible]es of **aut[illegible]e-writes…**
  - We **spent years patching** to a tool that **was never architected properly…**
  - We realize – too late - **we never understood the overall cost** of developing an in-house [illegible]ol.
  - We **got st[illegible]**.
  - We found [illegible] **situation** [illegible]out a huge investment…

We had enough!

… and then the automation guy got bored and left us with spaghetti code.

What is needed…

Small, localized, grass-root automation initiatives are welcome…

… as long as they stay small, localized and focused!

We need

□ *Alert Signals* to recognize the symptoms

□ *Counter Measures* to mitigate the impact

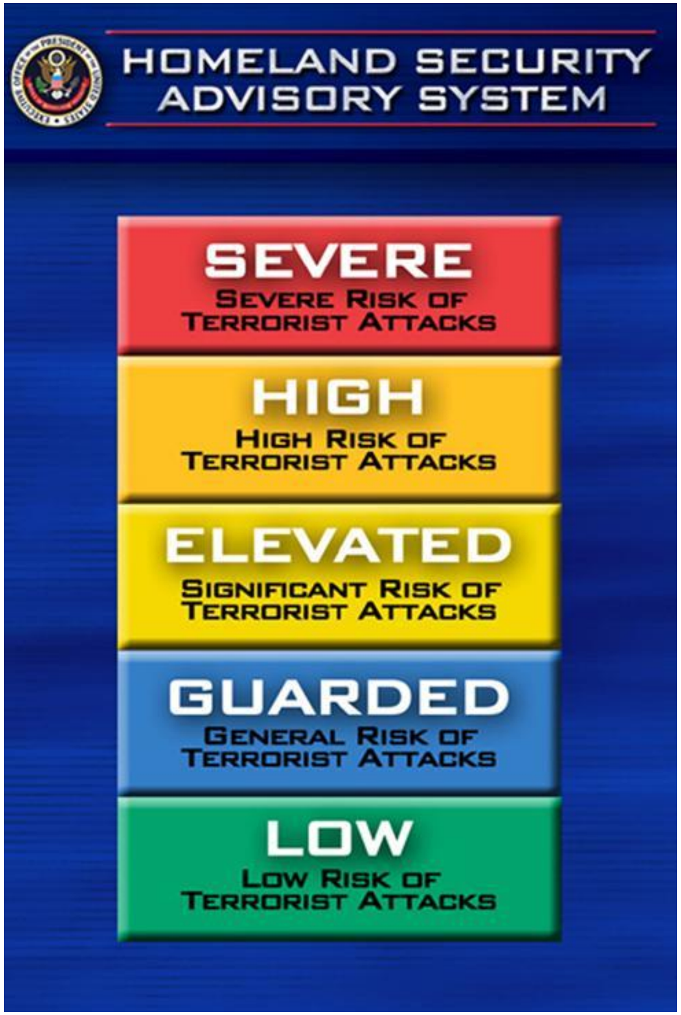
… of a runaway automation initiative

Alert Signals/indicators identified

The 5 stages can serve as Alert Signals

- □ Stage 1: Small, local, feature-centered
- □ Stage 2: Generalization
- □ Stage 3: Institutionalization and staffing
- □ Stage 4: Change of focus: Technology ⇨ Management
- □ Stage 5: Maintenance overload; Re-design

Alert Signals



HOMELAND SECURITY ADVISORY SYSTEM

SEVERE
SEVERE RISK OF TERRORIST ATTACKS

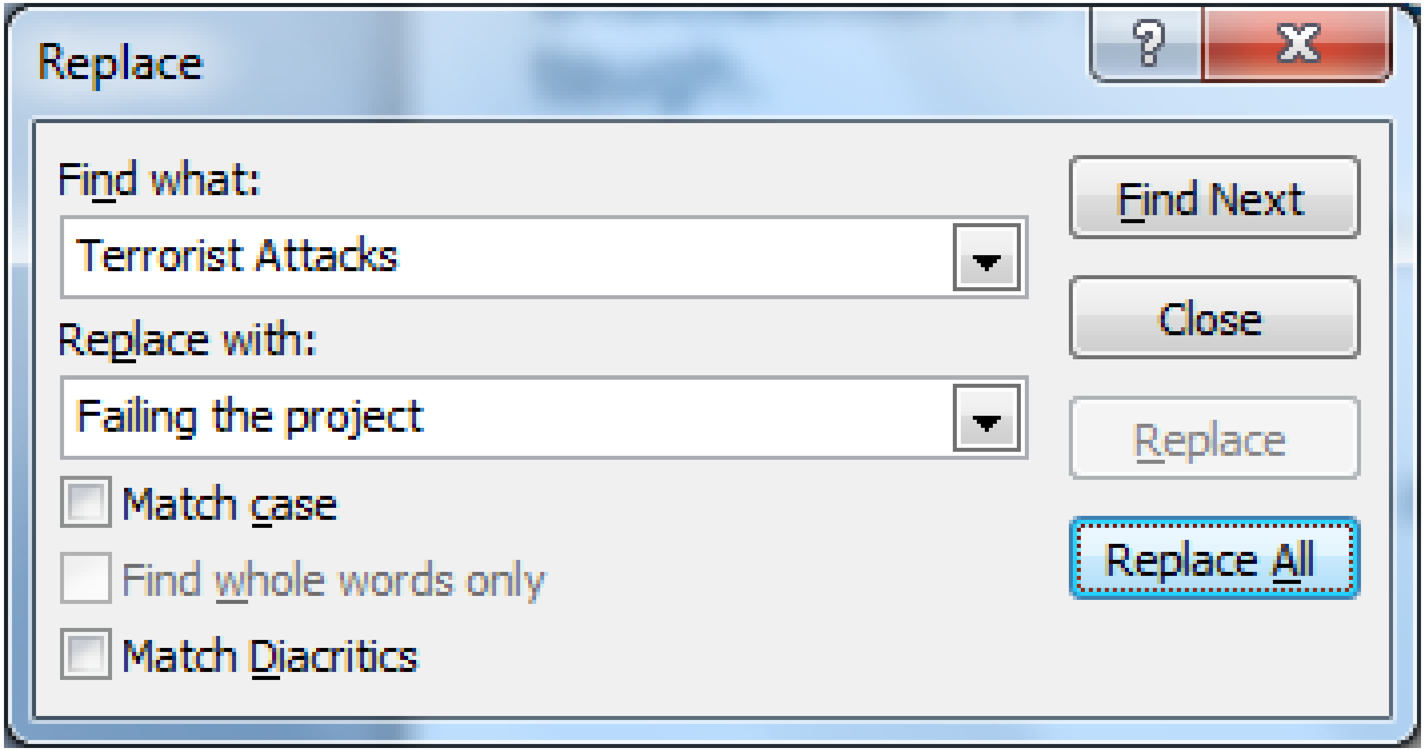
HIGH
HIGH RISK OF TERRORIST ATTACKS

ELEVATED
SIGNIFICANT RISK OF TERRORIST ATTACKS

GUARDED
GENERAL RISK OF TERRORIST ATTACKS

LOW
LOW RISK OF TERRORIST ATTACKS

Alert Signals



Find what: Terrorist Attacks

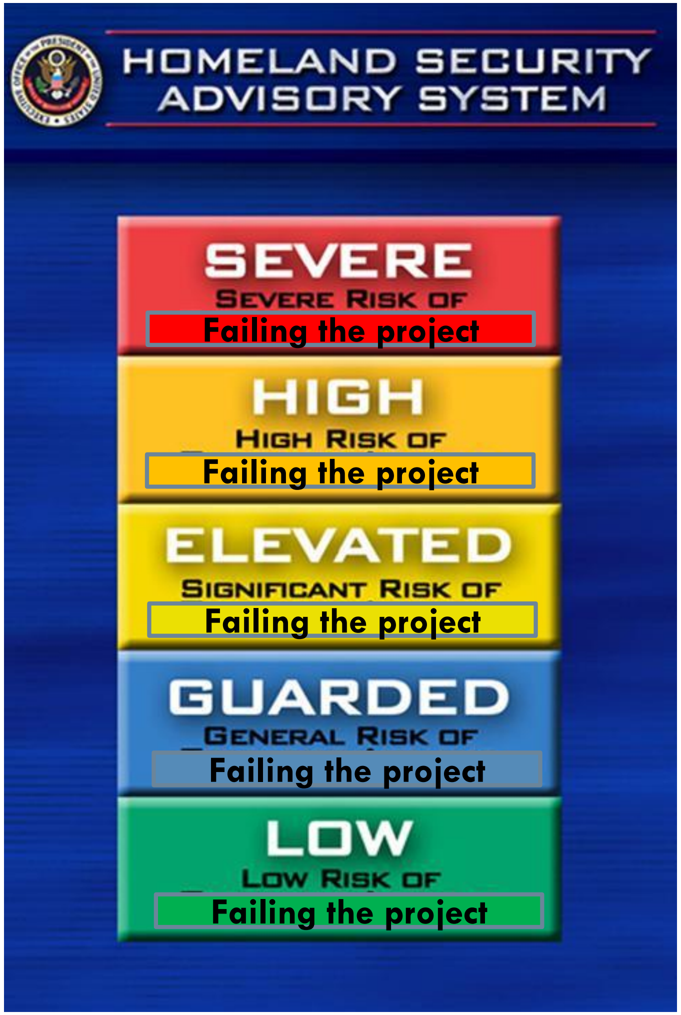
Replace with: Failing the project

□ Match case

□ Find whole words only

□ Match Diacritics

Alert Signals



HOMELAND SECURITY ADVISORY SYSTEM

SEVERE
SEVERE RISK OF
Failing the project

HIGH
HIGH RISK OF
Failing the project

ELEVATED
SIGNIFICANT RISK OF
Failing the project

GUARDED
GENERAL RISK OF
Failing the project

LOW
LOW RISK OF
Failing the project

Counter Measures

- On the next slides, you will find for each exit point/level:
  - Questions to ask – identify where we are?
  - Situation at this stage – symptoms identified
  - Suggested Counter Measures - aggregative

Alert signs

1. A Tool is built locally on a small scale

LOW

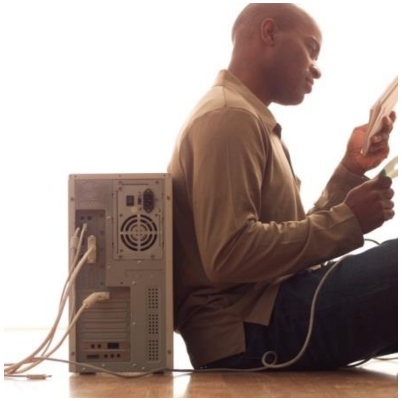
- **Questions:**
  - Single feature?
  - Clear ROI for the task it automates?
  - Single user – dev + run?

Counter Measures

1. A Tool is built locally on a small scale

LOW

- Ensure the following… and relax:
  - Source Code control
  - Documentation
    - User manual
    - High level design
  - Unit testing the tool will be done, as it is localized…



Alert signs

GUARDED

2. Requests are placed to enhance the tool and make it more generic (generalization)

- Additional features?
- **Multiple users?**
- Automation web site / wiki?
- >25% of the tester's time? – 1+ week/month, 1.5+ days/week
- HR-related?

Alert Signs

GUARDED

2. Requests are placed to enhance the tool and make it more generic

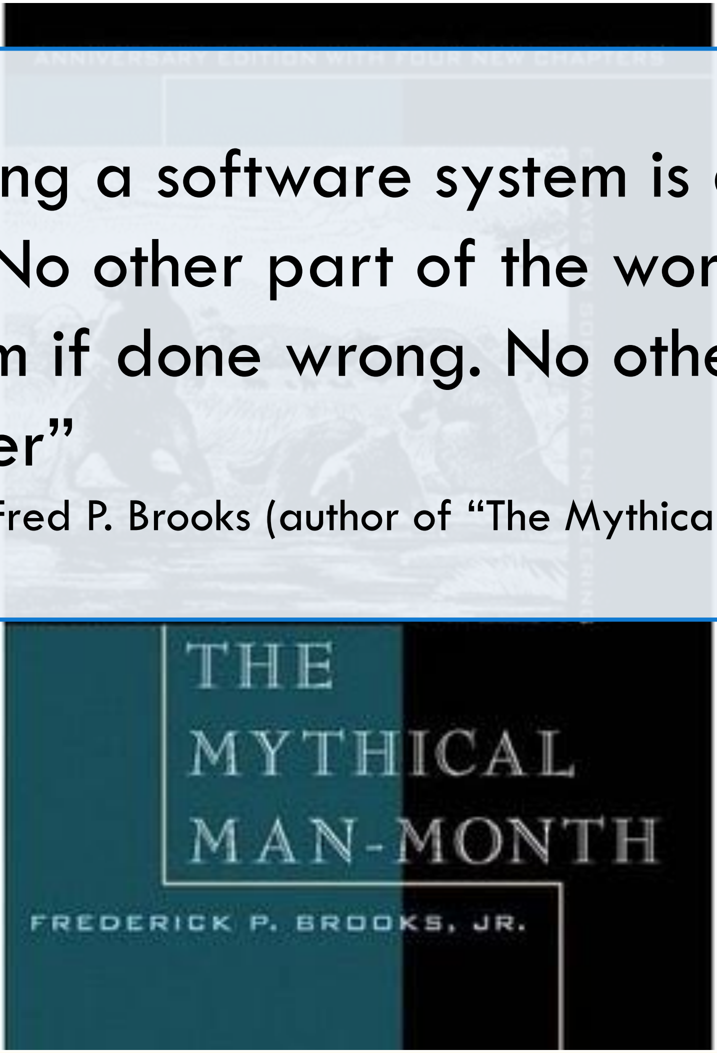
- **Situation at this stage**
  - The tool is still on small scale:
    - No generic test-case management capabilities
    - Implements mostly core-business or core-technology
  - Can’t be bought outside – **we want this tool!**
  - But we don’t want it to expand and become to much generic…

Counter Measures

GUARDED

“The hardest part of building a software system is **deciding precisely what to build…** No other part of the work so cripples the resulting system if done wrong. No other part is more difficult to rectify later”

- Fred P. Brooks (author of “The Mythical Man-Month”)



THE MYTHICAL MAN-MONTH

FREDERICK P. BROOKS, JR.

Counter Measures

2. Requests are placed to enhance the tool and make it more generic

GUARDED

- Stage 1 measures – SC, documentation…
- Automation Strategy
- Architecture and design
- Standardize tool development
- No GUI automation… (buy/open source…)
- Lightweight PM
  - Version control
  - Scope control
  - Bugs & Requests database

Alert Signs

3. Technical owner (de-facto) spends most of the time on automation with the tool

ELEVATED

- TA development skills start to be a limitation…
- Requests for additional “heads”?
- Automation issues resolve F2F?
- Too many automation tasks…
- Missing timelines?
- Tool-related delays in testing?
- Design related arguments are dragged on forever..

Alert Signs

3. Technical owner (de-facto) spends most of the time on automation with the tool

ELEVATED

- “we need a platform…”
- ROI?
- Knowledge transfer (sit with the guy for 2 hours)? Documentation?
- Defect reporting established…
- Key words:
    - “Prioritization”, “Tool Owner”, “Framework”; “Infrastructure”, “Roll back”, “Bug fix release”

Alert Signs

3. Technical owner (de-facto) spends most of the time on automation with the tool

ELEVATED

Probable situation at this stage:

- The tool supports a number of features, used by a number of people
- Some libraries exist… no clear strategy and design direction
- If you only caught this now, probably:
  - No documentation
  - No proper Development processes
  - No proper configuration management (folder…separate CM system…etc.)

Counter Measures

3. Technical owner (de-facto) spends most of the time on automation with the tool

ELEVATED

Counter Measures

3. Technical owner (de-facto) spends most of the time on automation with the tool

ELEVATED

- Hold everything + Complete step 2 mitigation list
- Cancel – efforts to building a test-management tool
- Stop – any effort of test automation, until finishing the next steps!
- Initiate **tool evaluation project**:
  - Getting the requirements part is difficult; Put a senior person on it; give it priority and the needed time (at least 2-3 months!)

Counter Measures

3. Technical owner (de-facto) spends most of the time on automation with the tool

ELEVATED

- Discuss and define "automation strategy"
  - "buy VS build"
  - "core technology VS generic test-management needs"
- Evaluate ROI again…build a business case
- Demand a proper CM: tool, process, environment
- Use same bug reporting tool as for production
- Include project management tasks
- Establish design and other development documentation rules and processes on TA

Counter Measures

3. Technical owner (de-facto) spends most of his time on automation with the tool

ELEVATED

- Integrate reporting to your commercial management tool
- **Enhancing skills** - Send your automation person to relevant computer-science and software development training
  - You need a developer – not a tester !
  - More people = Hire programmers
  - Transition phase = tester into developer…
- Acceptance testing

Counter Measures

ELEVATED

3. Technical owner (de-facto) spends most of his time on automation with the tool

Metrics suggestions

- Content and Progress – areas of TA, Coverage of risks & requirements
- *Automation framework quality*
  - Number of false fails
  - Test results; Bug trends
- ROI
  - Number of runs
  - Number of bugs found by Automation
  - Invested effort by type (new, maintenance, rewrite)

Counter Measures

ELEVATED

3. Technical owner (de-facto) spends most of his time on automation with the tool

Metrics suggestions – continue…

- Effort invested in automation – new scripts, maintenance scripts, things that were totally changed (waste)
- Bugs in automation vs. bugs in product (with severity High+)
- # bugs found with scripts – regression vs. new code
- # runs in various development stages
- # down time due to automation and the % of that from the whole effort
- More…

Alert Signs

4. Tool focus degrades – less core functionality is implemented

HIGH

- **Questions:**
  - How many features you have developed in-house vs. exist in commercial or open-source tools? [low.. medium..high…]
  - How many features are not related directly to the core functionality and technology of the product (%)? [a lot…]
  - How much time of your automators is spent on maintenance? [>25%…]
  - Did you do (or are you discussing) a "technical enhancement release"? [Yes…]

Alert Signs – continue…

HIGH

4. Tool focus degrades – less core functionality is implemented

Questions:

- TA people are needed to analyze results?
- Too many log files? Folders? Manual operations?
- Key words:
  - “test suite / cycle generation”
  - “robustness enhancement”
  - “setup issues”

Alert Signs

HIGH

4. Tool focus degrades – less core functionality is implemented

- **The situation at this stage**
  - Lots of code was written, strongly influencing a “buy VS build” decision
  - People start to relay on the tool for their day to day testing…
  - The tool’s robustness is low. New releases are painful !
  - Confidence of test automation reliability is starting to degrade

Counter Measures

4. Tool focus degrades – less core functionality is implemented

HIGH

- **Hold up everything!!!**
- Complete stage 2+3 mitigations
- Create test automation/Tool clear roadmap
- Re-architect
  - Core technology and functionality vs. non-core
  - Solid infrastructure
- Stabilize tool code related to Core/technology – architecture, modularity, maintenance, bugs
- Build a clear test plan for test automation tool/environment, and a release plan and process

Counter Measures

HIGH

4. Tool focus degrades – less core functionality is implemented

Build (VS Buy) if…

- ☑ Competitive edge
- ☑ Existing expertize
- ☑ Core competency
- ☑ Cheaper; Faster
- ☑ Good use of resources
- ☑ Acceptable risk
- ☑ Long term support

Main source: Allen Eskelin – http://www.informit.com/articles/article.aspx?p=21775

Alert Signs

5. Tool re-design is unavoidable…

SEVERE

Questions:

- Many chefs effect…
- Maintenance & logistics overload?
- The framework is not modular?
- New features vs. bug fixes → bug fixes…
- “because the automation is like that… I cannot do this…”
- Loss of credibility?
  - “Report a bug only when it was reproduced manually”
  - “Never mind the automation; I’ll just run it manually; it takes less time”

Alert Signs

5. Tool re-design is unavoidable…

SEVERE

Questions:

- Stage 1 initiatives?
- Key words
  - “Did it fail in manual test?”
  - “Architecture limitation”
  - “refactoring”; “redesign”
  - “…let me write a small program…”

Alert Signs

5. Tool re-design is unavoidable…

SEVERE

Questions:

- Does the tool suffer from **performance problems**?
- **Unplanned delays in test-cycles** due to test-automation tool failures?
- **Late-nights and weekends** on automation trends up?

The Test Automation project is in Critical state ! ! !

Counter Measures

SEVERE

5. Tool re-design is unavoidable…

- Continue… ☹
- Give up problematic areas – go manual on specific areas, for TA to recover…
- Partial return to Stage 1
- "We value Robustness over New Features…" - define acceptance criteria for new scripts
- Prepare for re-design – properly!

Desperate
TESTERS

Counter Measures

SEVERE

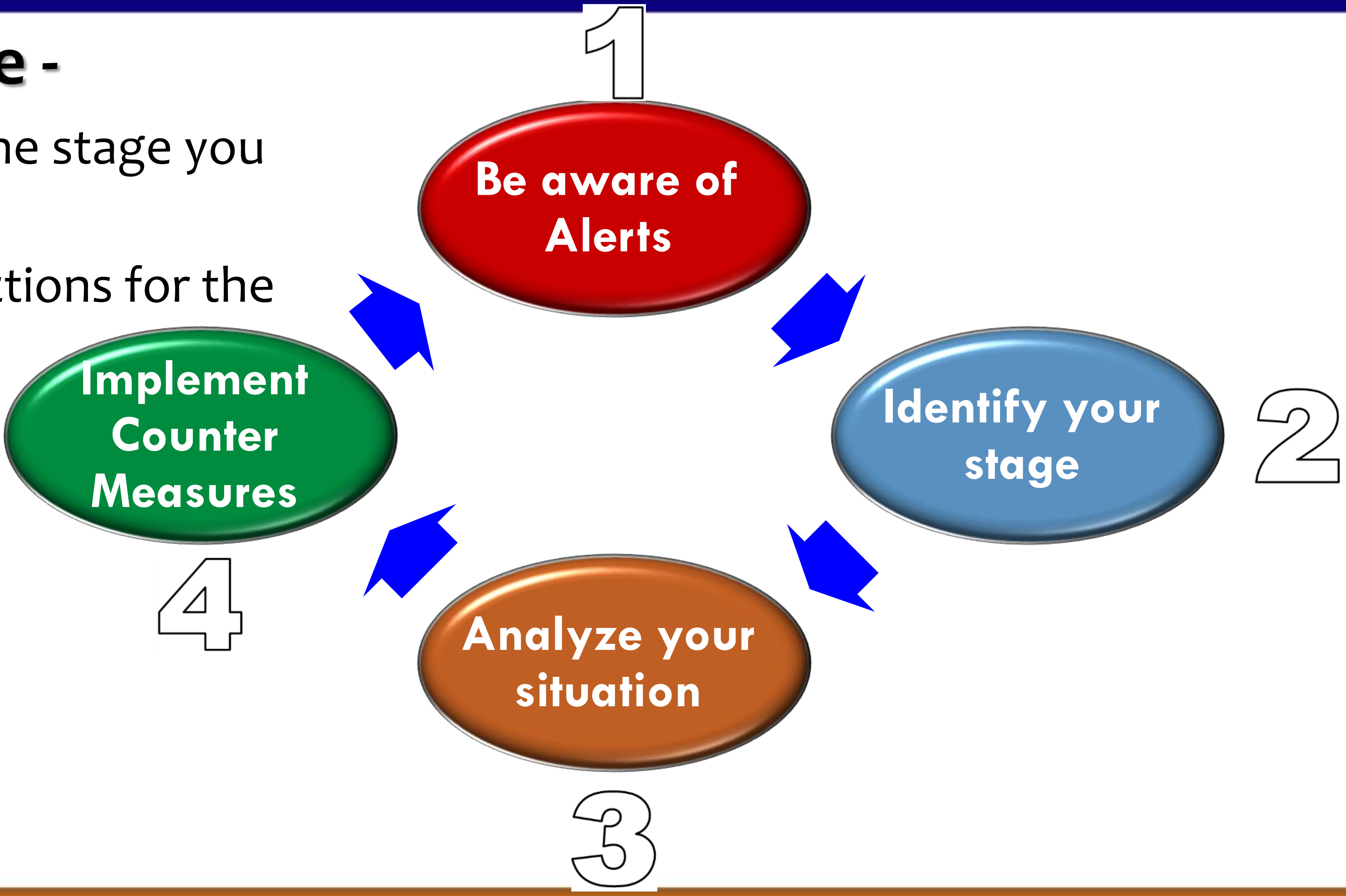
5. Tool re-design is unavoidable…

- Minimal support to keep the tool running ("alive")
  - Can't hold back the test cycles…
- Apply stages 2, 3, 4 mitigations !
- Improve coordination with Development
- Focus on ROI – no script is developed or executed if ROI is not clear

Desperate TESTERS

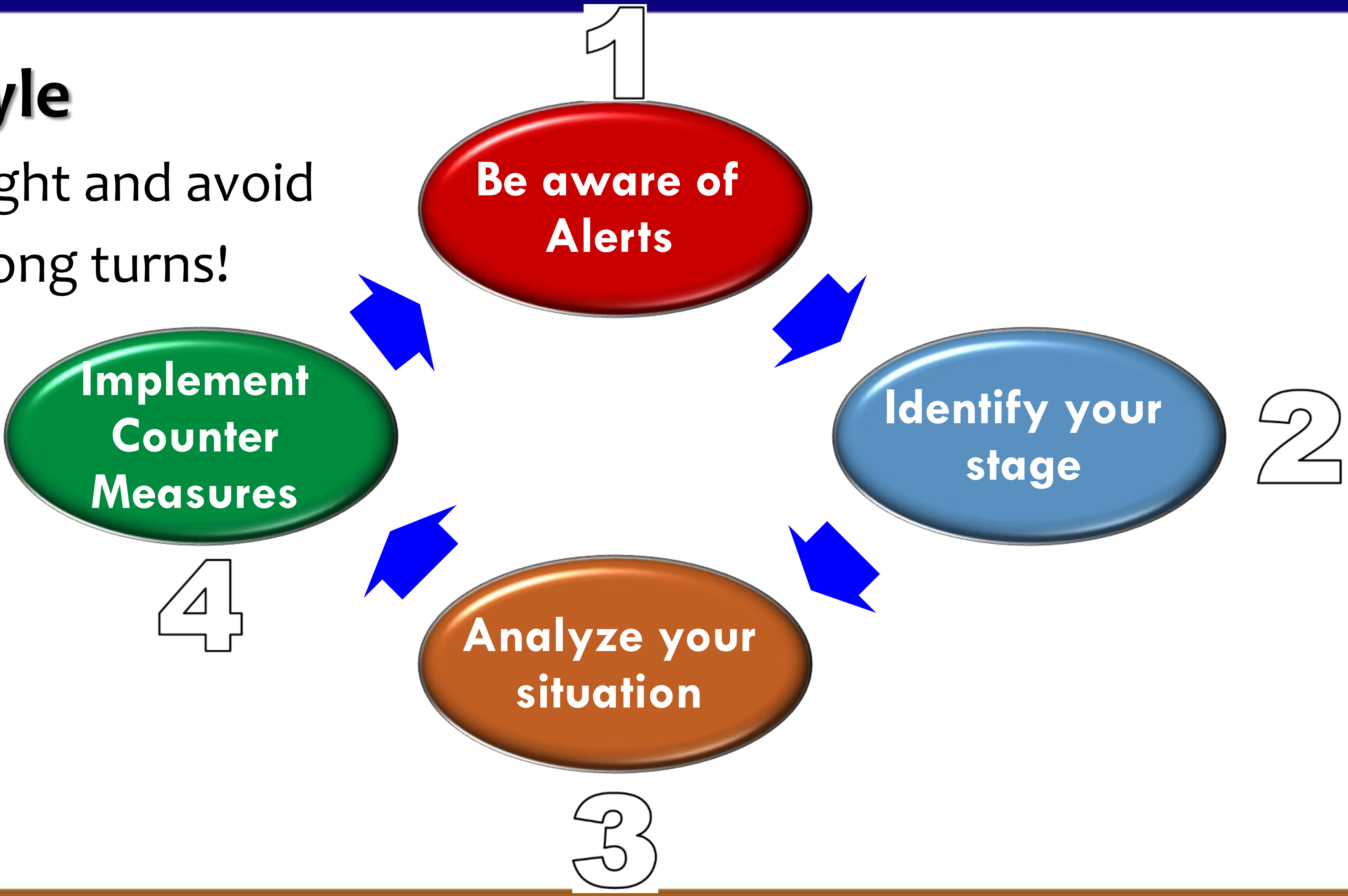
How to Use this information?

- **GPS style -**
  - Locate the stage you are at…
  - Get directions for the way out!



How to Use this information?

- **Map Style**
  - Start right and avoid the wrong turns!



Food for Thought…

- □ If my position is stage 3,4 or 5 - Why haven't I been to this workshop so far?
- □ How does what we have learned - related and implemented in Agile?
- □ Who should be setting up and managing the Test Automation – Dev or Test?
- □ Which management aspects are different in test automation (from testing)?
- □ Who else should be exposed to this information, knowledge and understanding so that I can get the budget and attention that I need?

Coming up

discussions…sharing ideas…learning…

Test Automation Failing Projects

4 groups, 4 flipcharts, choose a scribe/presenter.

Step 1:

Challenges and problems in those TA areas,

Areas:

Test Automation Tools & Framework

Test Automation Management

Test Automation Strategy & Methodology

Test Automation for NFT

Test Automation Failing Projects

Step 2:

Re-group near a TA topic you have solutions and experience in.

Discuss, write possible solutions to problems suggested by the 1st round.

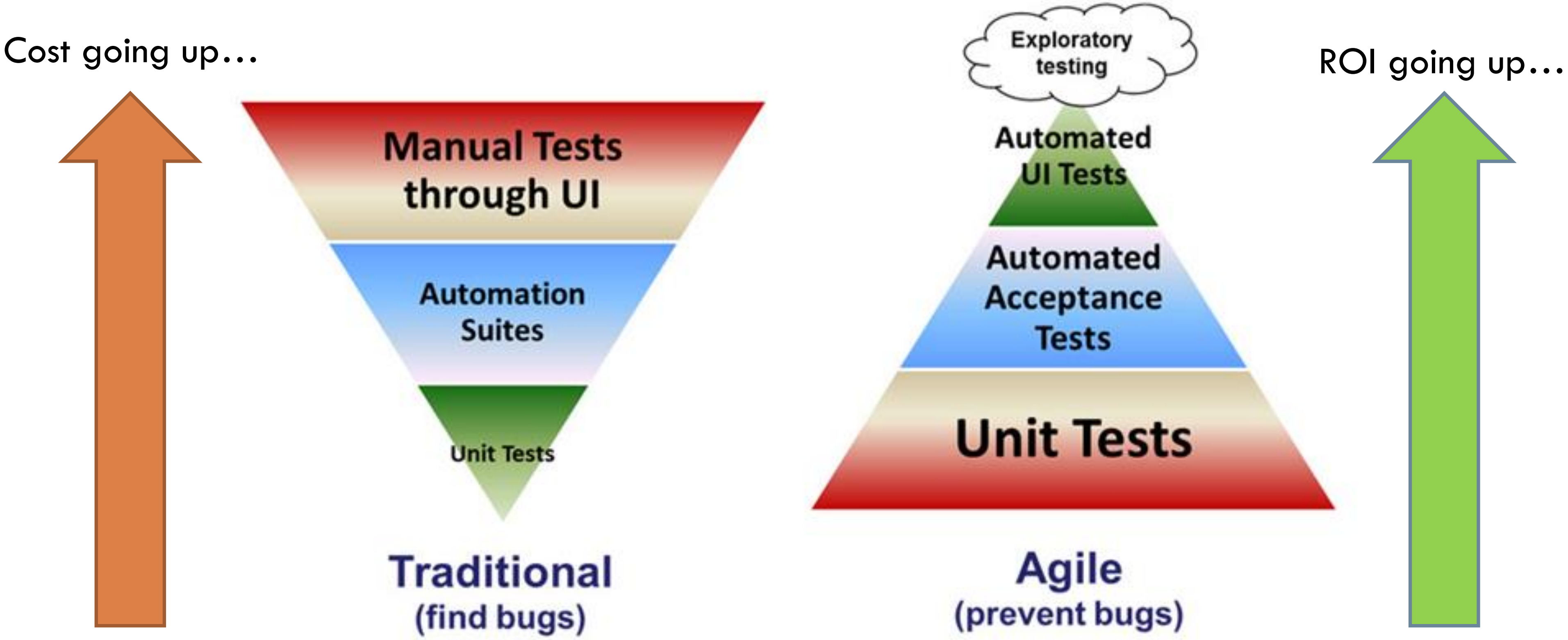
Solutions you are familiar with in those TA areas,

Bonus slides…

Test Automation under Agile…

- Shifting the test automation focus
- Emergency need for speed…smooth delivery and CI or CD – requires higher level of test automation
- Focus is shifted:
  - Most efforts on UT – best ROI
  - Medium size on integration/API
  - Low size on GUI/UI – lowest ROI
- See next…

Test Automation under Agile…



http://www.agilecoachjournal.com/index.php/2014-01-28/testing-2/the-agile-testing-pyramid/

Discussion

- In which solutions, automation is not such a good idea? Why?
- Where should test automation team report to – the testing group or the development group? Why?
- In Agile world, is the market demanding programming capabilities from testers? Should we all learn how to code? Why?

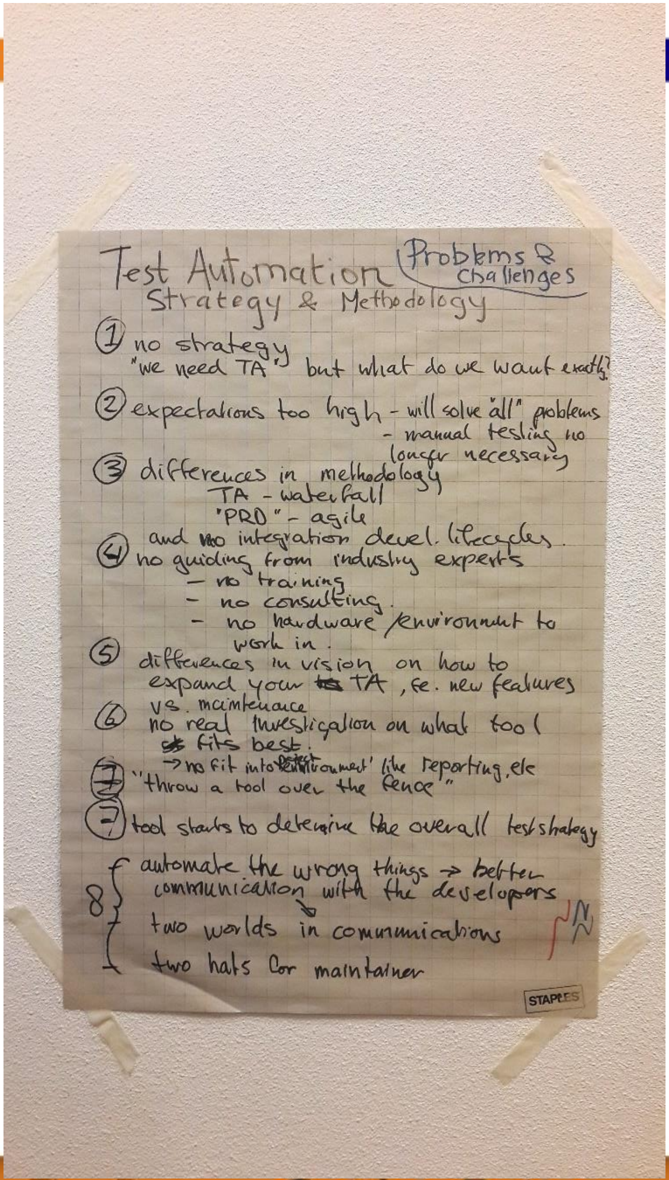
Summary of group discussion

- Below are the 4 group discussion summary hand writing as recorded during the workshop

- Enjoy!

Test Automation
Non Functional testing
Test Automation
Management
- bottom up vs
Top down approach
- MISSING INFORMATION
- NO ONE DECIDES
WHICH TOOL TO CHOOSE
Welkom bij NBC

Test automation strategy & methodology



Test Automation (Problems & challenges)
Strategy & Methodology

① no strategy
"we need TA" but what do we want exactly?

② expectations too high - will solve "all" problems
- manual testing no longer necessary

③ differences in methodology
TA - waterfall
"PRD" - agile
and ~~no~~ integration devel. lifecycles.

④ no guiding from industry expert's
- no training
- no consulting.
- no hardware /environment to work in.

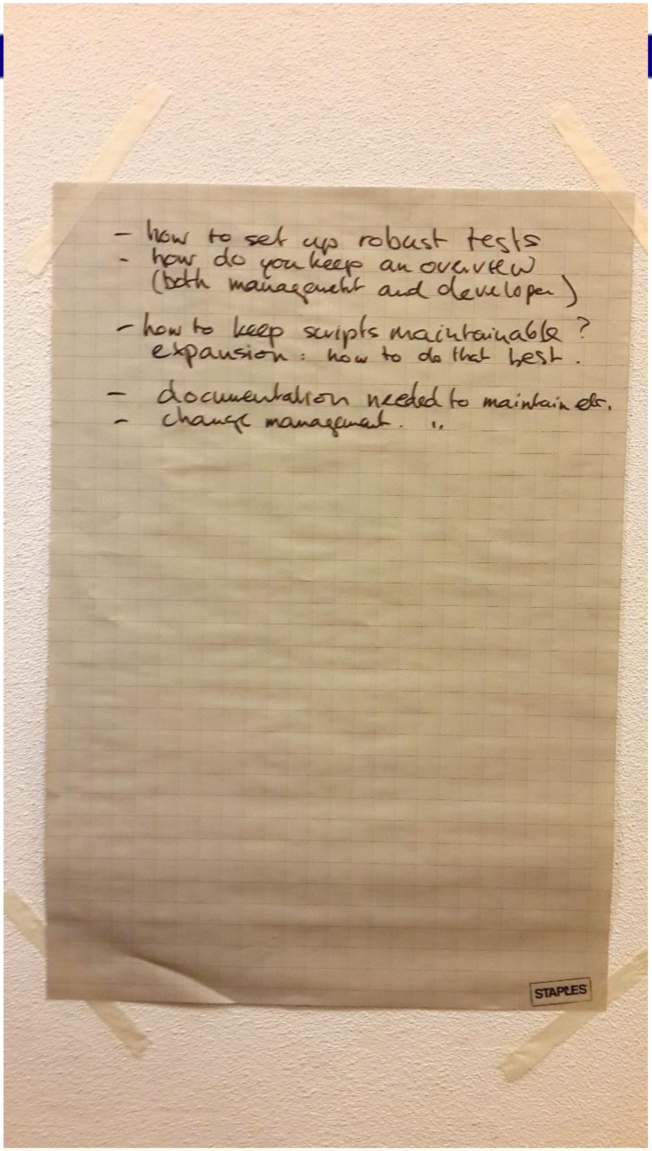
⑤ differences in vision on how to expand your ~~to~~ TA, fe. new features vs. maintenance

⑥ no real investigation on what tool ~~s~~ fits best.
→ no fit into project ~~environment~~' like reporting, etc

⑦ "throw a tool over the fence"

⑦ tool starts to determine the overall testshategy

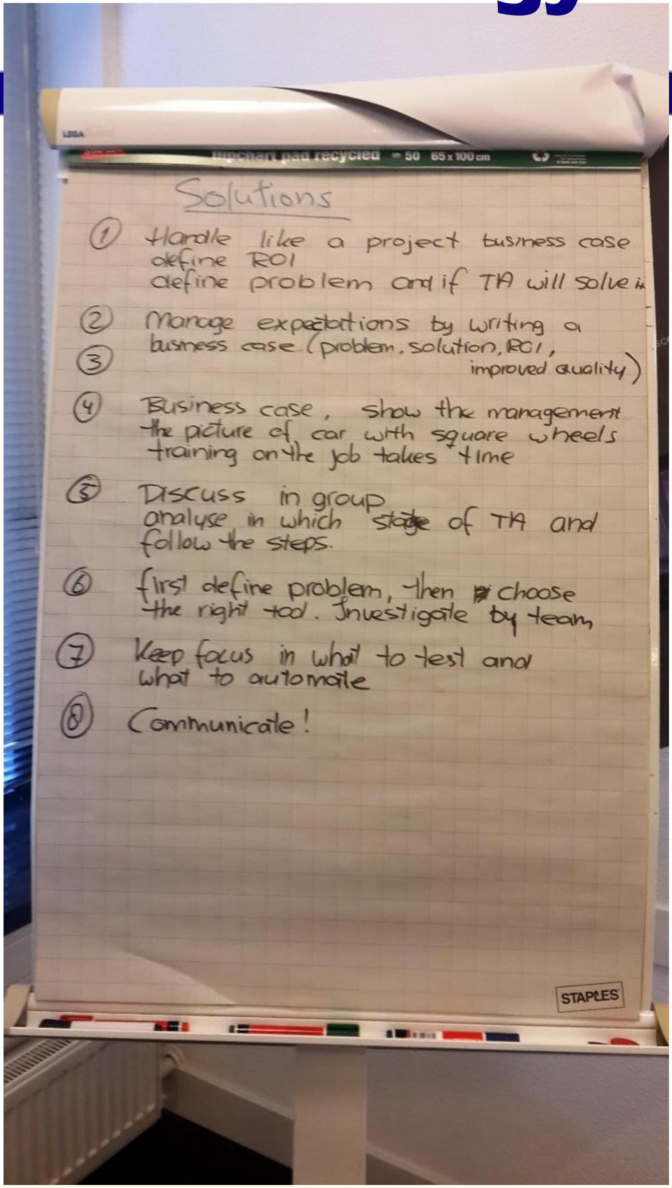
8 { automate the wrong things → better communication with the developers
two worlds in communications
two hats for maintainer



- how to set up robust tests
- how do you keep an overview (both management and developer)

- how to keep scripts maintainable? expansion: how to do that best.

- documentation needed to maintain etc.
- change management. "



Solutions

① Handle like a project business case
define ROI
define problem and if TA will solve it

② Manage expectations by writing a
③ business case (problem, solution, ROI, improved quality)

④ Business case, show the management the picture of car with square wheels
training on the job takes time

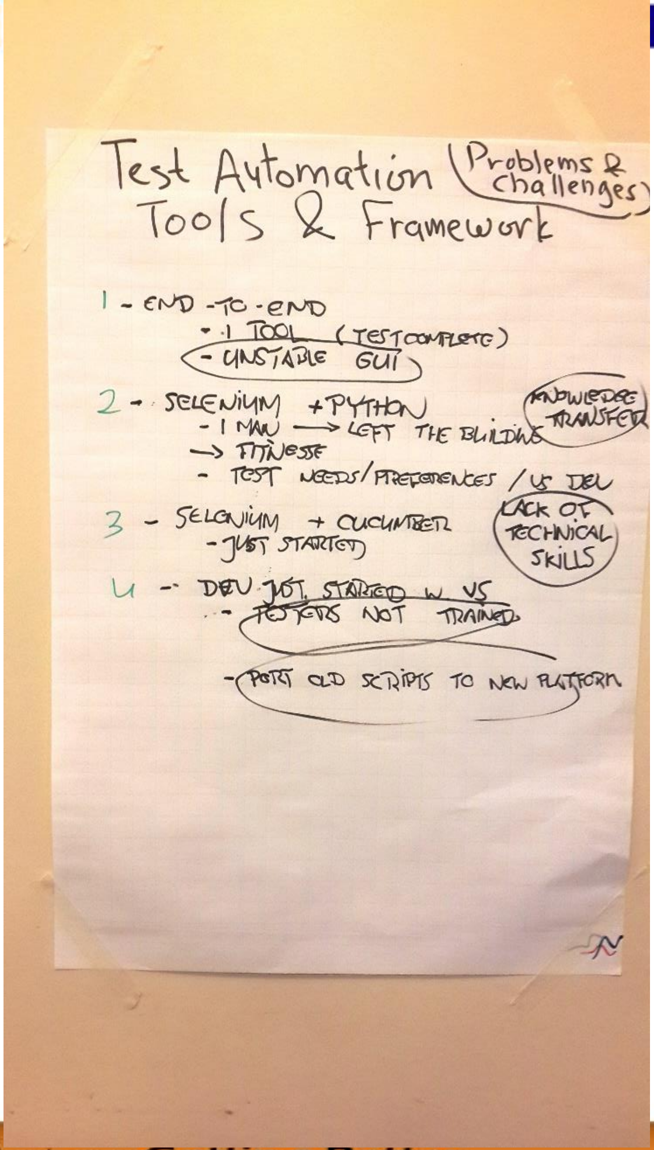
⑤ Discuss in group
analyse in which stage of TA and follow the steps.

⑥ first define problem, then choose the right tool. Investigate by team

⑦ Keep focus in what to test and what to automate

⑧ Communicate!

Test automation tools & framework



Test Automation (Problems & challenges)
Tools & Framework

1 - END-TO-END
- 1 TOOL (TESTCOMPLETE)
- UNSTABLE GUI

2 - SELENIUM + PYTHON
- 1 MAN → LEFT THE BUILDING
→ FITNESSE
- TEST NEEDS/PREFERENCES / VS DEV

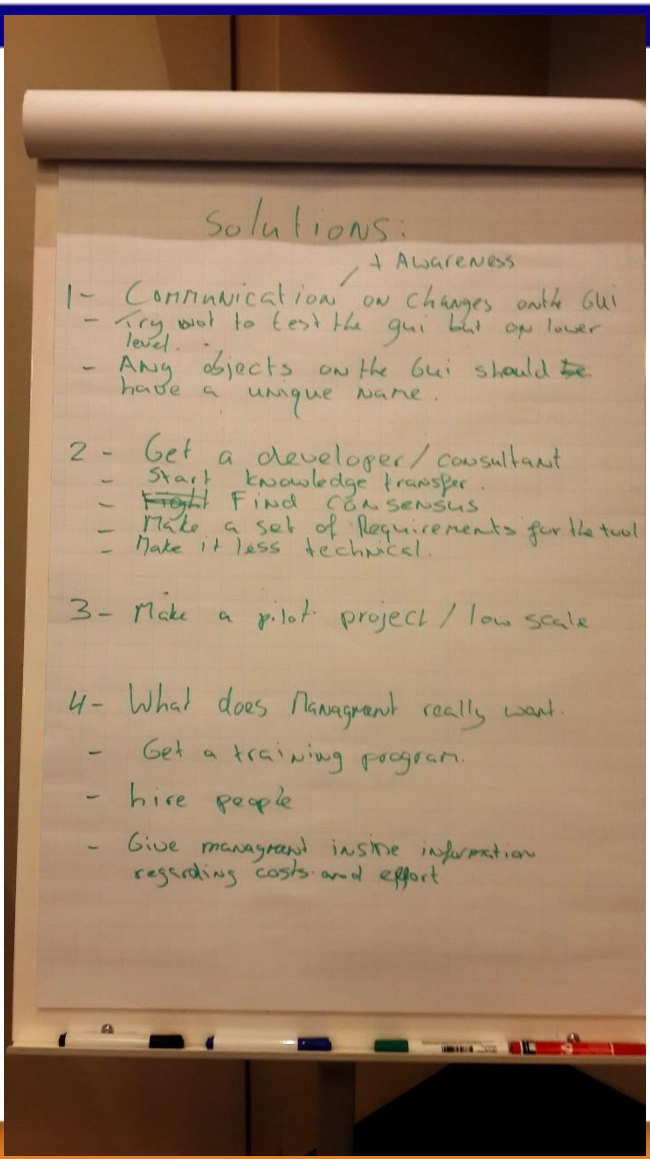
KNOWLEDGE TRANSFER

3 - SELENIUM + CUCUMBER
- JUST STARTED

LACK OF TECHNICAL SKILLS

4 - DEV JUST STARTED W. VS
- TESTERS NOT TRAINED

- PORT OLD SCRIPTS TO NEW PLATFORM



Solutions:

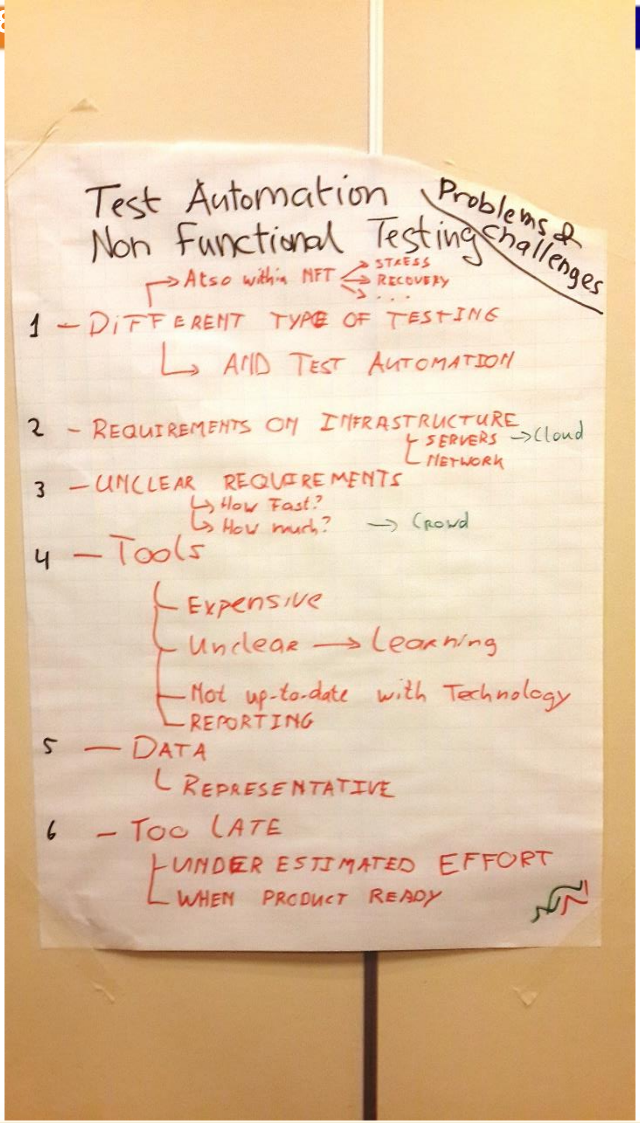
1- Communication + Awareness on changes on the Gui
- Try not to test the gui but on lower level.
- Any objects on the Gui should ~~be~~ have a unique name.

2 - Get a developer/consultant
- Start knowledge transfer.
- ~~Fight~~ Find consensus
- Make a set of Requirements for the tool
- Make it less technical.

3- Make a pilot project / low scale

4- What does Managment really want.
- Get a training program.
- hire people
- Give managment inside information regarding costs and effort

Test automation non-functional



Test Automation
Non Functional Testing

Problems & challenges

→ Also within NFT → STRESS → RECOVERY → ...

1 – DIFFERENT TYPE OF TESTING
└→ AND TEST AUTOMATION

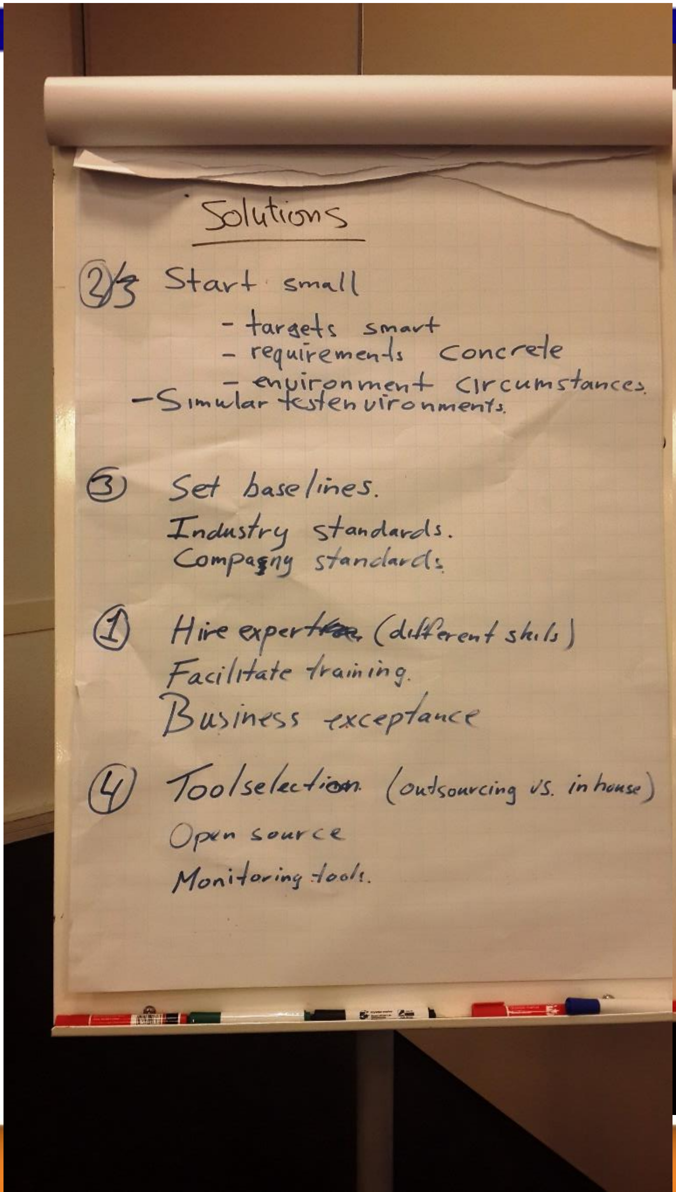
2 – REQUIREMENTS ON INFRASTRUCTURE
└ SERVERS → Cloud
└ NETWORK

3 – UNCLEAR REQUIREMENTS
↳ How Fast?
↳ How much? → Crowd

4 – Tools
- Expensive
- Unclear → Learning
- Not up-to-date with Technology
- REPORTING

5 – DATA
└ REPRESENTATIVE

6 – TOO LATE
- UNDER ESTIMATED EFFORT
- WHEN PRODUCT READY



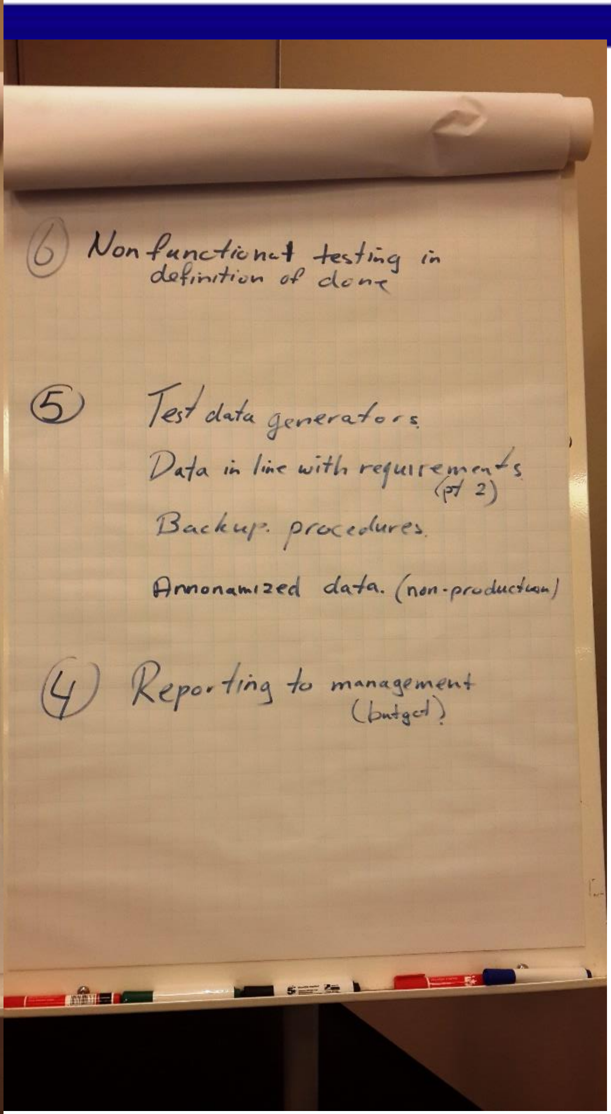
Solutions

2/3 Start small
- targets smart
- requirements concrete
- environment circumstances.
- Simular testenvironments.

3 Set baselines.
Industry standards.
Compagny standards.

1 Hire ~~expertise~~ (different skils)
Facilitate training.
Business exceptance

4 Toolselection. (outsourcing vs. in house)
Open source
Monitoring tools.

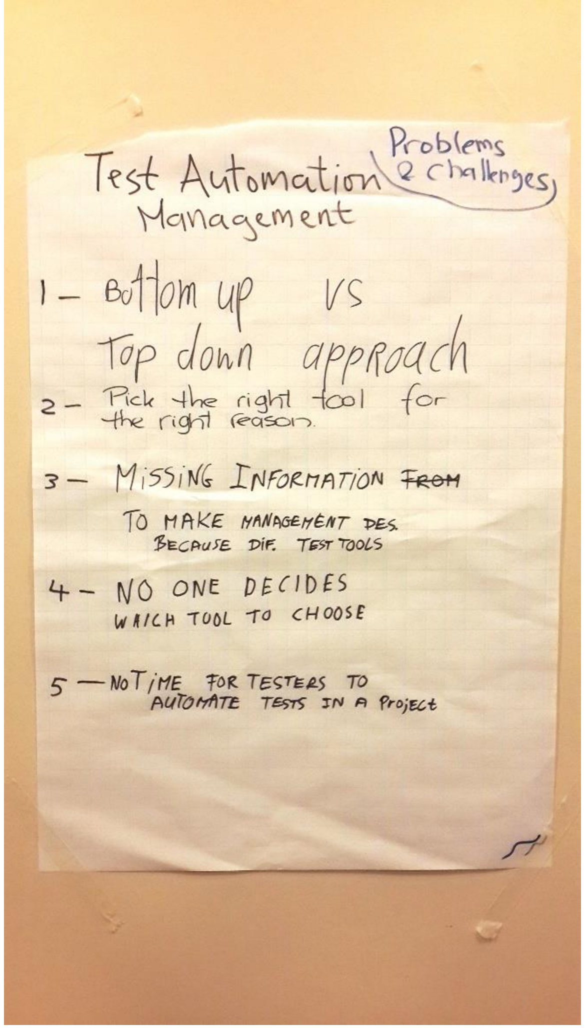


6 Non functional testing in definition of done

5 Test data generators
Data in line with requirements (pt 2)
Backup procedures.
Annonamized data. (non-production)

4 Reporting to management (budget)

Test automation management



Test Automation Management

Problems & challenges

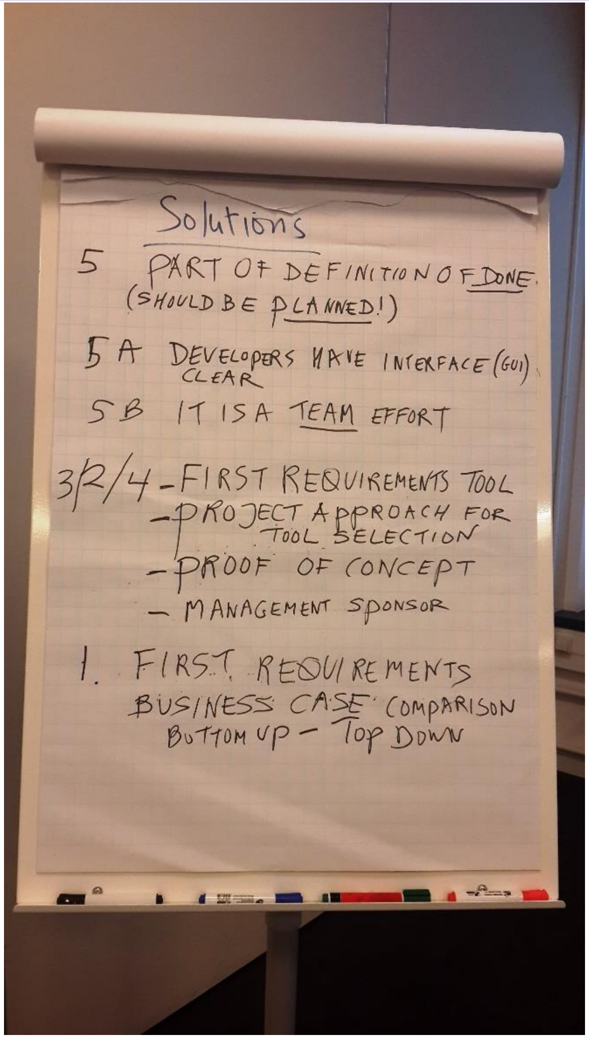
1 – Bottom up VS Top down approach

2 – Pick the right tool for the right reason.

3 – MISSING INFORMATION ~~FROM~~ TO MAKE MANAGEMENT DES. BECAUSE DIF. TEST TOOLS

4 – NO ONE DECIDES WHICH TOOL TO CHOOSE

5 – NO TIME FOR TESTERS TO AUTOMATE TESTS IN A Project



Solutions

5 PART OF DEFINITION OF DONE. (SHOULD BE PLANNED!)

5 A DEVELOPERS HAVE INTERFACE (GUI) CLEAR

5 B IT IS A TEAM EFFORT

3/2/4 – FIRST REQUIREMENTS TOOL
- PROJECT APPROACH FOR TOOL SELECTION
- PROOF OF CONCEPT
- MANAGEMENT SPONSOR

1. FIRST REQUIREMENTS
BUSINESS CASE COMPARISON
BOTTOM UP – TOP DOWN

Inspiration…

IF YOU NEVER GIVE UP, THEN YOU HAVE ALREADY SUCCEEDED…

YOU ARE BETTER THAN YOU HAVE EVER BEEN BEFORE.

THANK YOU!

ALON LINETZKI, ALONL@SIGIST.ORG.IL

http://alumpofclayinafrica.blogspot.nl/2012/08/about-2-years-ago-god-impressed-on-our.html

Best Testing
The best are getting better

THANK YOU!

The Road to Hell and Back

Patterns in Test Automation project failure - and Recovery

The Best are Getting Better

Alon Linetzki ⊕ www.Best-Testing.com ✉ alonl@sigist.org.il ☎ +972 (0)54 4848533