Hyper-automation of requirements analysis
Helping Software Architects

**Agenda**
1. What is hyper-automation of requirements analysis?
2. What does it look like?
3. What can it do for you as an architect?
4. Why does it matter?
Presenter
Colin Hammond M.Eng MBCS CFPS
30 Years student of IT

colin.hammond@scopemaster.com
Over 28 years of seeing repeated pattern of Root Cause Problems

Large Projects
- Requirements poor quality
- Requirements incomplete
- Requirements not sized

Terrible Record
- 71% cancelled or challenged (~$180Bn)*
- 19% cancelled
- It’s actually worse than this

*Standish report 2020, USA)
Root Cause of Defects Found in Production

Most Activity
- Unit testing
- Systems testing
- Functional testing
- End to end testing
- Acceptance testing

1,000 FP Application
Source: Capers Jones
Applied Software Measurement, third edition
Hyper-automation what is it?

Hyper-automation

• Using technology to bring orders of magnitude improvements to hitherto manual work.
A typical agile user story:

**Add Delivery Details**

**As a ...** Site visitor  
**I want ...** Add my delivery addressss  
**So that ...** I can receive my goods

Acceptance/Test Criteria ...

I can click pencil to enter my zip code and full home address

**“Who & What”**

- Using NLP and more. Automate sound software practices.
- Flexible, any phraseology, any taxonomy
Requirements Quality Really Matters

Poor user stories lead to waste and amplified rework

1 user story word

On average

125 code “words”

12 words

1500 SLOC

Based on analysis of over 100,000 user stories by ScopeMaster
Intelligent Analysis - including automated functional sizing

1. Analyses ANY phraseology

2. Detects functional steps

3. Detects Objects & intent

4. Determines data movements

5. Estimates functional size

Place an order at the table

Validate that the device is permitted. Validate permissions for the waiter. Then as a waiter I can insert the order

Perform thousands of context-aware tests and analysis steps on every story in just 1-5 seconds
Case study to compare SP vs COSMIC Function Points

**Story points vs actual effort**

$R^2 = 0.33$

**CFP vs actual effort**

$R^2 = 0.97$

CFP a Reliable Predictor of Effort

Fixing bugs in a later phase is VERY expensive

Fix early is least expensive
Architects’ Concerns on larger projects

Things Many Architects worry about, may be hard to spot and hard to assess:

1. Quality (objectives->req->arch->design->code->test->data)
2. Complexity
3. Size
4. Coupling (between modules) & cohesion (within modules)
5. Traceability - requirements to objectives and code to req.s
6. PM related questions - risk, schedule, cost
## Analysing a user story

**Administrator can modify a user’s profile**

*As an administrator first validate superuser permissions, then I can modify a user profile*

<table>
<thead>
<tr>
<th>Edit</th>
<th>Functions</th>
<th>Quality</th>
<th>History</th>
<th>Tests</th>
<th>Related</th>
<th>BDD</th>
<th>Debug</th>
</tr>
</thead>
</table>

### Short title

Administrator can modify a user’s profile

### Functional Requirement

As an administrator first validate superuser permissions, then I can modify a user profile

### Benefits

Benefits

### Notes

All except password.
Demonstrates more than one functional step within a use story.

Last updated: 19 days 20 hours 38 minutes 58 seconds ago
Last analysed: 22 days 20 hours 59 minutes 57 seconds ago
Last estimated: 19 days 20 hours 38 minutes 58 seconds ago

Reference: US05

Requirement type: Functional
10 Quality Attributes for Better User Stories

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clear</td>
<td>✔️</td>
</tr>
<tr>
<td>Concise</td>
<td>✔️</td>
</tr>
<tr>
<td>User-oriented</td>
<td>✔️</td>
</tr>
<tr>
<td>Testable</td>
<td>✔️</td>
</tr>
<tr>
<td>Measurable</td>
<td>✔️</td>
</tr>
<tr>
<td>Consistent</td>
<td>✔️</td>
</tr>
<tr>
<td>Complete</td>
<td>✔️</td>
</tr>
<tr>
<td>Unique</td>
<td>✔️</td>
</tr>
<tr>
<td>Valuable</td>
<td></td>
</tr>
<tr>
<td>Design-free</td>
<td>✔️</td>
</tr>
</tbody>
</table>

ScopeMaster helps in 9 out of 10 of these categories. Overall ScopeMaster is able to find and help you fix 50-65% of all requirements defects.
Analysing and cross correlating stories

Press Play
A set of user stories

lost in words?
Use Case Models Generated Automatically

Explore your user stories visually

Stimulates critical thinking:

Exposes Complexity & Coupling
Automated Diagrams

Automated Visualisations
Promotes critical thinking
Validate and Verify - visually
Finds and helps Fix Problems - FAST!

Data maintenance analysis

Find and fix potential missing and duplicate requirements. Each maintained object of interest usually has one Create, Read, Update and Delete operations.

<table>
<thead>
<tr>
<th>Object (19) confirm</th>
<th>Create (8)</th>
<th>Read (12)</th>
<th>Update (3)</th>
<th>Delete (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>address</td>
<td>Missing+</td>
<td>address lookup</td>
<td>Missing+</td>
<td>Missing+</td>
</tr>
<tr>
<td>cart</td>
<td>Missing+</td>
<td>Display cart</td>
<td>Missing+</td>
<td>Missing+</td>
</tr>
<tr>
<td>cart_item</td>
<td>add to cart</td>
<td>display cart_item</td>
<td>Missing+</td>
<td>Missing+</td>
</tr>
<tr>
<td>delivery_address</td>
<td>add delivery address</td>
<td>display delivery_address</td>
<td>Missing+</td>
<td>Missing+</td>
</tr>
</tbody>
</table>
### Tracing code to requirements - Suggested Class diagram

#### product
- `maintain_products(C)()`: Maintain products
- `create product`
- `read product`
- `search_products()`
  - `read product`
  - `Search (Alternative Bad story example)`
- `update product`
  - `Maintain products`
- `delete product`
  - `Maintain products`

#### product category
- `add_product_categories()`
- `create product category`
- `add product categories`

#### profile
- `authenticate_profile()`
- `update profile` (Login)

#### search
- `add_search()`
  - `create search`
    - `Search (Alternative Bad story example)`
- `search_display()`
  - `read search`
    - `Search (Alternative Bad story example)`

#### search term
- `add_search_term()`
  - `create search term`
    - `Search (Alternative Bad story example)`

#### superuser permission
- `validate_superuser_permissions()`
  - `read superuser permission`
  - Administrator can modify a user's profile
  - [FINAL]
Large Projects / Transformations

- Poor requirements cause quality and rework problems & delays
- Agile doesn’t scale easily - artful architectural separation
- Size matters
- Valid measurement - greater transparency and predictability

Large project benefit the most
Benefits

Requires no set up, just import your requirements and press "analyse"

1. Better User Stories
   - Find and fix problems fast

2. Instant Estimates
   - ISO standard sizing
   - Estimates functional size
   - Finds missing requirements!

3. Auto Test Creation
   - Baseline tests generated
   - Perfect traceability
   - Huge time-saver

Automated:
- Interprets stories
- Tests stories ~700 per story.
- Builds a data dictionary

Benefits:
- Expose problems
- Helps fix before coding
- Reduce rework
- Valid estimates
- Better informed decisions
- Save on test prep time
- Ensure coverage
Conclusion

Key takeaways

• Hyper-automation of requirements analysis exists.
• Brings scrutiny and insight to requirements, reducing waste
• “extreme shift-left testing”
• Built on sound proven methods
• Non-trivial benefits

colin.hammond@scopemaster.com
https://www.scopemaster.com
https://cosmic-sizing.org
https://www.amazon.co.uk/Capers-Jones/e/B000APTHHW?
Portfolio Overview

Portfolio Size:
- 8,129 Requirements in 90 apps
- 29,271 Cosmic Function Points

Project size:
- C-REG MM IMPORT TEST
  - reimport of C-REG MM IMPORT TEST 1
  - OWNER access
  - 100% MEASURABLE
  - 165 CFP
  - 22 of 22

- Library 2 (as user stories)
  - Same as library 2 but written as user stories
  - OWNER access
  - 88% MEASURABLE
  - 7 of 8
  - 2.5

- A big one
  - test of auto generated requirements
  - OWNER access
  - 64% MEASURABLE
  - 833 CFP
  - 102 of 160

Quality:

ScopeMaster QUALITY SCORE: 3.5
- Ambiguous: 0
- Pot. Missing: 88
- Pot. Duplicated: 9

DEFECTS:
- Ambiguous: 1
- Pot. Missing: 19
- Pot. Duplicated: 23

Total Cost: $3,000

Total Defects: 821
Case study to compare SP vs CFP

### Story points vs actual effort

\[ R^2 = 0.33 \]

### CFP vs actual effort

\[ R^2 = 0.97 \]

**Conclusion:**

CFP is a better predictor of effort than story points.

About Functional Size

Valid
Standard
Non-gameable
Suitable for agile
Suitable for contracts
Ideal for creating estimates
Manage: Scope, Resources, Schedule and Quality.

Average value of knowing size: 10-40% of total budget.
Sizing software

Functional Size Metrics on Software Projects

- **Agile Story Points**
  - Very Flawed:
    - Not Valid
    - Inconsistent
    - Easy to game
  - Flawed:
    - Not Valid
    - Inconsistent
    - Easy to game
  - Good:
    - ISO Standard
    - Consistent
    - User stories insufficient
    - Not ideal for embedded
  - Best:
    - ISO Standard
    - Incomplete OK
    - Principle-based
    - Automated
    - US. GAO Recommended

- **SLOC**
- **RICEFW**
- **IFPUG FP**
COSMIC Function Points - The best way to measure software work

\[ \sum E, X, R, W = CFP \]

- Persistent storage
- Read
- Write
- Entry
- Exit
- Application being sized
- Other interfacing App(s) or devices

Appropriate & Valid
Consistent
Mature and stable
Language independent
Methodology independent
Technology Independent
Suitable for all S/W
Open source / free
ISO Standard
How ScopeMaster helps reduce Architecture Risk

*Architecture attributes that reduce risk:*

**Coupling**
ScopeMaster highlights data coupling between requirements

**Cohesion**
ScopeMaster highlights data and requirement relationships
Missing, duplicates identification exposes cohesion
Functional size is also a good indication of cohesion

**Complexity**
ScopeMaster exposing size (an indicator of complexity)
Ambiguity exposure leads to lower complexity
Software Tools that help you write better software

### Automated Analysis

- ScopeMaster

### Requirements Modelling

- MATLAB & SIMULINK
- bluep...<br>
- ENTERPRISE ARCHITECT

### Requirements Capture and project management

- Microsoft Excel
- Jira Software
- Pivotal Tracker
- Rally Software
- Visual Studio
- Azure DevOps

### Coding & Testing

- CAST
- MICRO FOCUS
- SYNOPSYS
- Semmle
- CODE CLIMATE
- JET BRAINS
- CODEBEAT
- eggplant
- sonarqube
- SMARTBEAR
- GitHub