

AUTOMATED
SOFTWARE TESTING
AND QUALITY



Hosted by  **PARASOFT**

Manual Testing isn't Quality

Break the vicious cycle of manual software testing and deliver better software faster!

Presented by Nicholas Cioran
Head of Applications Research and Advisory Services

Info-Tech Research Group Inc. is a global leader in providing IT research and advice. Info-Tech's products and services combine actionable insight and relevant advice with ready-to-use tools and templates that cover the full spectrum of IT concerns.

© 1997-2020 Info-Tech Research Group Inc.

INFO~TECH
RESEARCH GROUP

**10 PRINT HELLO WORLD!
RUN
HELLO WORLD!**



Technology teams have become the enablers of growth for most organizations

In a *software eats everything else* world, demands on technology teams are growing. Management expects them to...



Source: The Developer Coefficient, 2018.

Analyst Perspective

Digital Delivery is Automated



A modern SDLC is the *heartbeat of the digital software delivery umbrella* of Agile, DevOps, and Product-centric methods enabled by an integrated toolchain. However, the latest technology has only compounded, at speed, the oldest problem in Information Technology; namely *what to automate and how to automate it*.

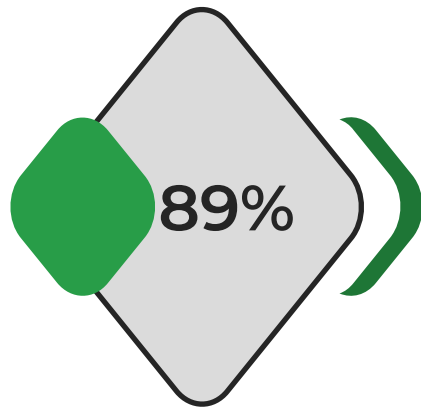
Cole Cioran

Head of Applications Research and Advisory Services
Info-Tech Research Group

Is automated testing the exception?

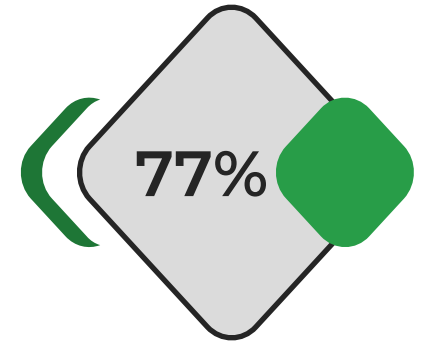
Automated testing has significant inroads in modern testing practices

Your stakeholders expect high product quality and delivery throughput and we've turned to automated testing to support that need.



89% of organizations are doing some form of automation. In particular, 77% of organizations report that they have automated and/or scripted their functional or regression tests.¹

Most organizations (77% of those surveyed) included test automation and scripting tasks as a tester's responsibility.¹



Automating your routine and avoiding error-prone and labour intensive manual test cases enable your testers to focus on higher-value activities.

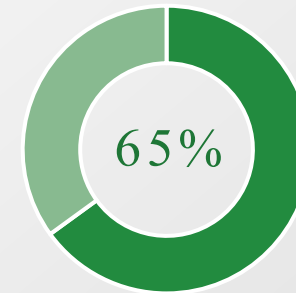
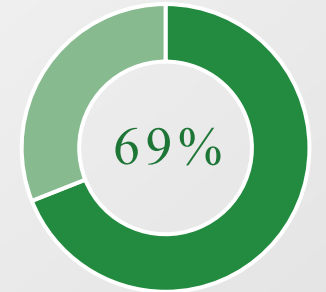
¹Source: PractiTest, 2020



Image Source: [wuyouyuan CC BY-SA 3.0](#)

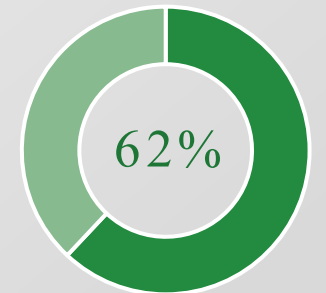
Automated testing has important benefits

Of organizations saw better control and transparency of test activities



Of organizations saw reduction of test cycle time.

Of organizations saw reductions of test costs.



Source: Sogeti, 2021

Lack of discipline reduces automated test coverage of requirements

Practitioners must avoid these pitfalls when building a disciplined automated testing practice.



Avoid Local Optimization

Increased throughput will not be possible if practitioners only look at pieces of the testing pipeline. Local optimization can create downstream bottlenecks for teams who cannot accommodate increased loads and aggressive expectations.



Automated Testing Is an Afterthought

Automated testing is often very far from the business value that stakeholders see. It's an easy candidate to cut when development is under pressure.

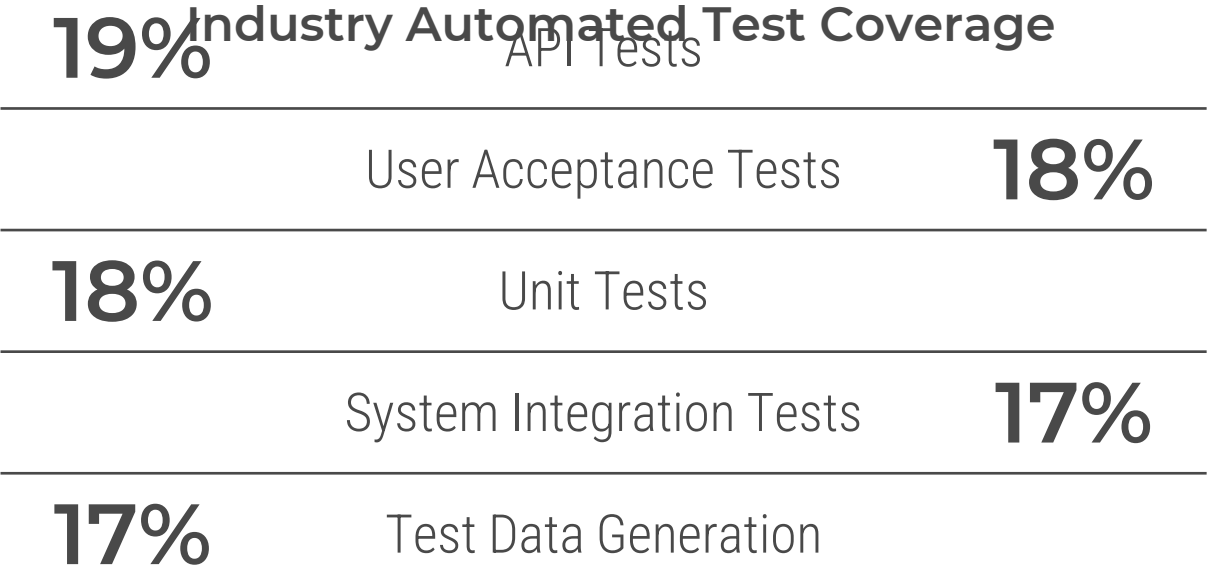


Automated Tests Are Not Regularly Validated and Maintained

Automated tests are coded and deployed like any other business tool. However, they are not treated the same, risking long-term viability and low ROI.

Effort alone will not create test coverage.

Despite 63% of respondents saying they have enough time to build and maintain their automated tests, test coverage remains low.



Source: Sogeti, 2021



The State of the Art?

8%

Of committed work gets delivered in a reasonable timeframe

20%

Of features are cancelled during project delivery

35%

Of product teams will not have capacity for new work for one year

85%

Of product teams underinvest in security, compliance and technical debt

95%

Of product teams have no visibility into end-to-end SDLC wait times

Source: Tasktop Viz® Cross-Industry Flow Metrics
Insights from the Global 2000, 2021, n = 787

But what about the Agile Manifesto?

*We value individuals
and interactions over
processes and tools...*



An organization's people are its most valuable capital assets, with technology teams being essential in this category.

They understand the business deeply and are de facto drivers of innovation. To ensure their value keeps increasing, training them in the latest tools, technologies, and processes must be a strategic priority.

1

Reimagine your SDLC.

Recognize that automation without optimization reinforces the status quo and is a recipe for long-term pain. Do it right the first time.

2

Stop writing testing (and other) documents.

Documents perpetuate low efficiency legacy SDLC methods and make it hard for software teams and stakeholders to understand whether they are delivering the right thing.

3

Target optimization and automation where it matters most.

Target tooling on the least effective parts of the SDLC to reduce bottlenecks and create end to end visibility into the effectiveness and efficiency of the SDLC.

Optimate!

“The first rule of any technology used in a business is that automation applied to an efficient operation will magnify the efficiency. The second is that automation applied to an inefficient operation will magnify the inefficiency.” – Bill Gates

Automation should be applied to a lean SDLC where activities that add no value are eliminated or streamlined.

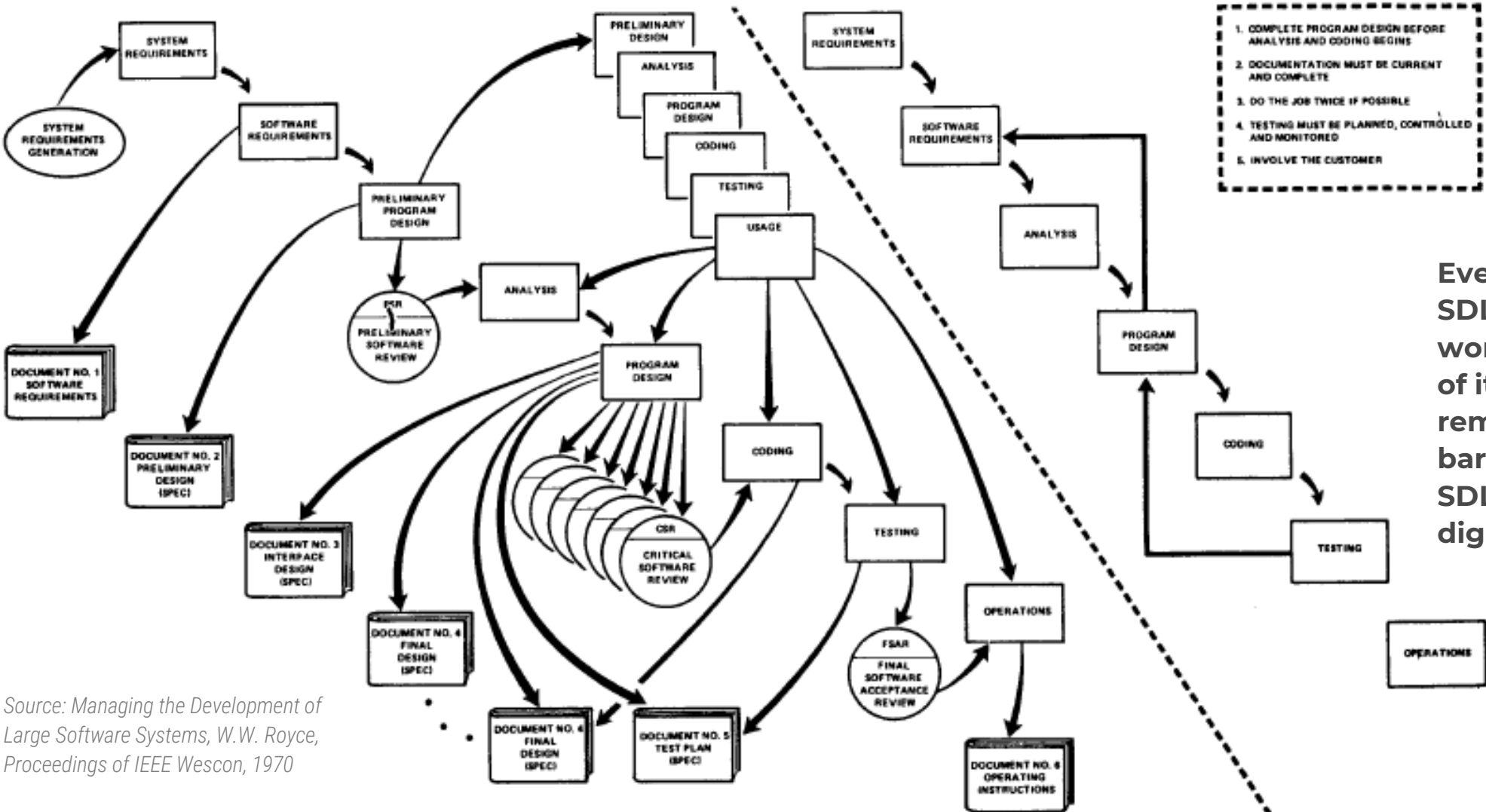
Recovered capacity from automation must first be applied to learn and employ new methods and technologies to accelerate delivery.

Info-Tech Insight

OPTIMATE your SDLC

Optimization and automation go hand in hand. Optimization is the cake, and automation is the icing on top. They are also an ongoing exercise. If left in status quo, technical debt and inefficiencies will inevitably creep back in.

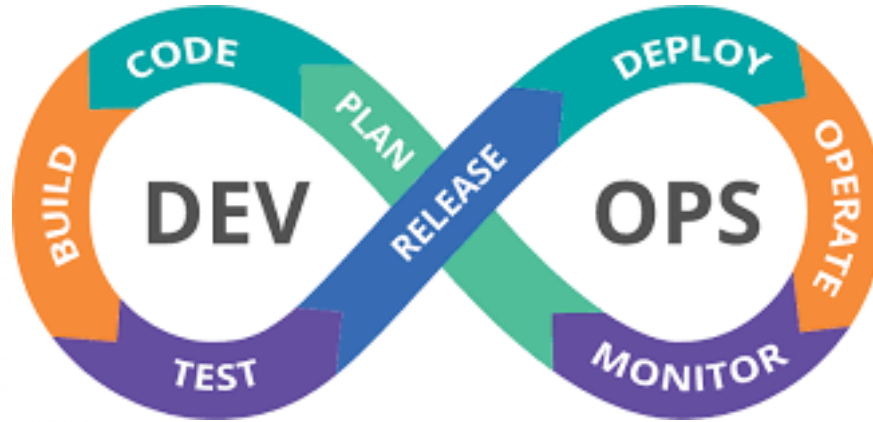
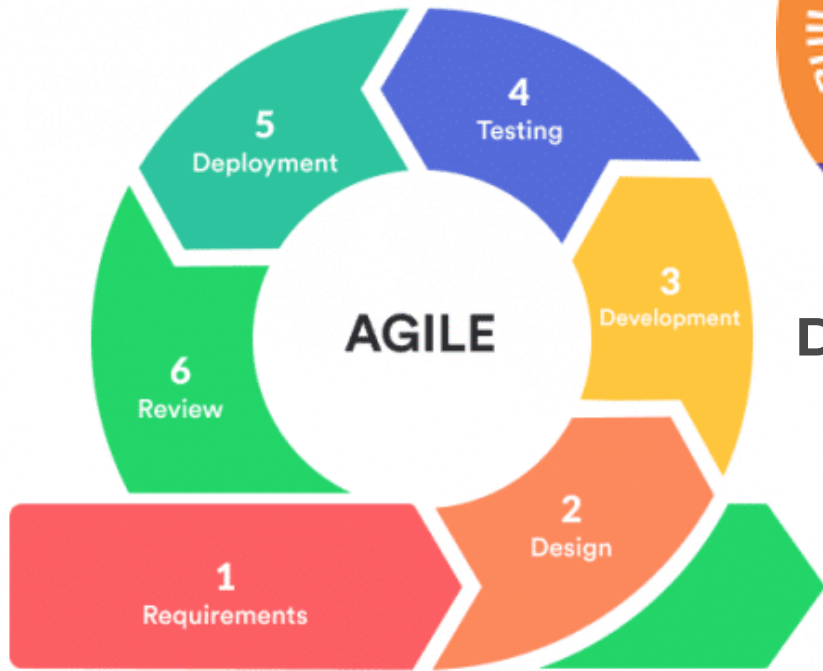
Step 1: Re-imagine your SDLC



Even when we break the SDLC down into smaller work items our conception of it as a series of stages remains the greatest barrier to modernizing an SDLC and creating a true digital delivery engine.

Source: *Managing the Development of Large Software Systems*, W.W. Royce, Proceedings of IEEE Wescon, 1970

What about Agile, DevOps, and BizDevOps?



Despite having had a solution since 1960s, visualizations of evolving concepts in software delivery highlight that we still haven't broken the fundamental conceptualization of the SDLC as a linear process.



Start by putting quality in the center

Doing the
Right
Things



A quality-centric
SDLC drives a **73%**
improvement in
throughput

Doing
Things
Right



Info-Tech Insight

Quality Assurance is the number one driver of throughput in software delivery, outstripping every other IT factor. As a result, it is critical to ensure quality is built into not just the software itself, but all of the activities required to deliver working software efficiently and effectively.

Change your models to change minds

Legacy approaches lead to legacy thinking.

Swim Lanes

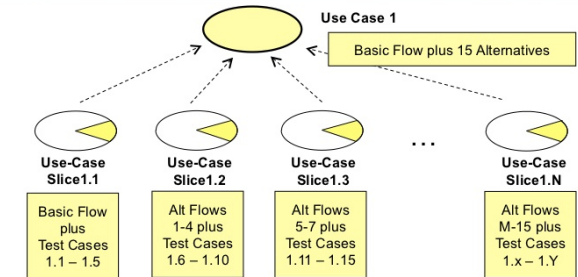


Swim lanes do not readily support the design of human-computer interactions

or

Use Cases

A use case may generate several slices covering all its functionality ...



The Use-Case Slices split the use case up into a number of smaller, separately deliverable parts

IVAR JACOBSON INTERNATIONAL

16

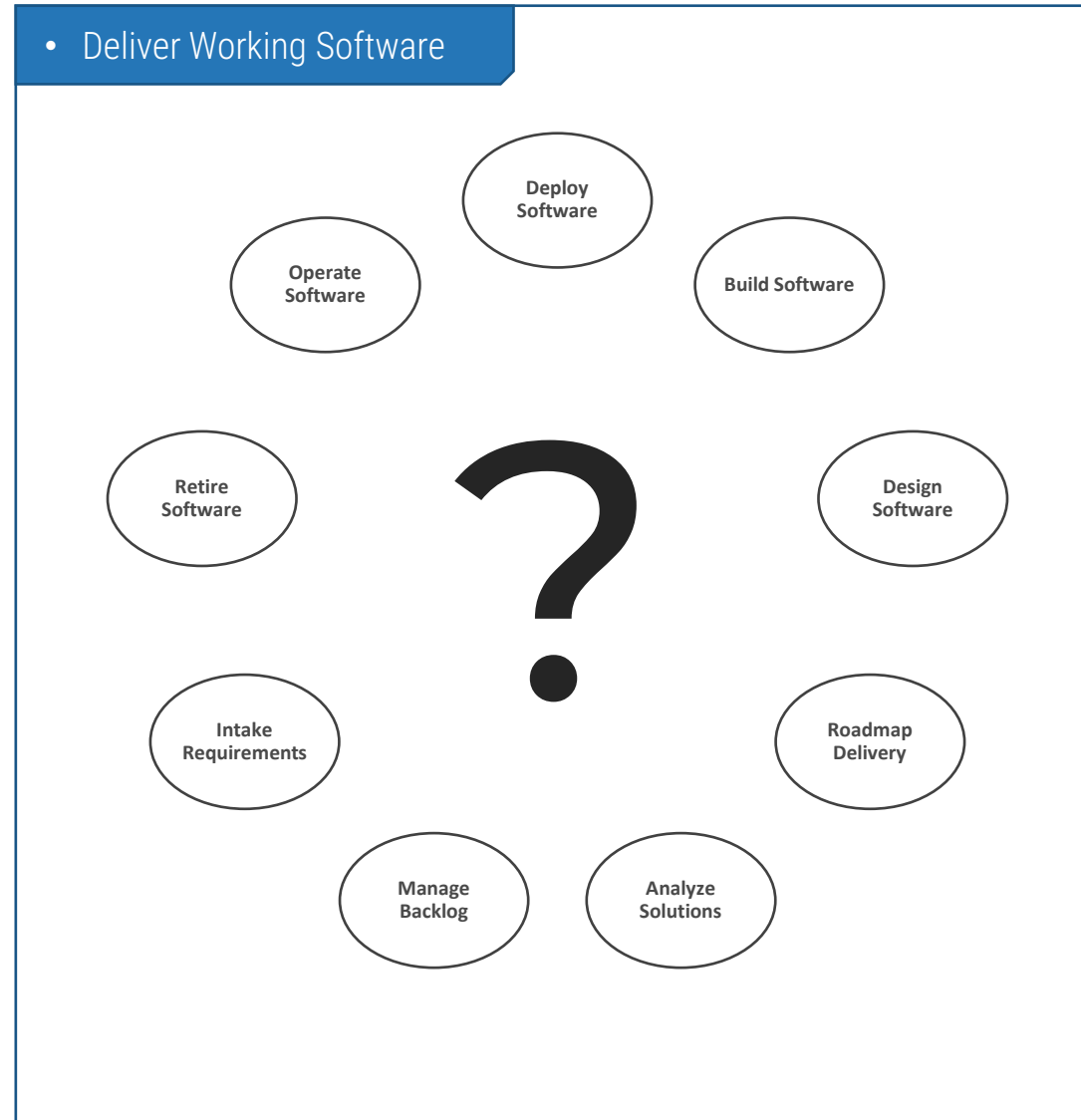
THE SMARTER WAY
Copyright © 2011 Ivar Jacobson International SA. All rights reserved

Use case models put users in the center of human-computer interactions

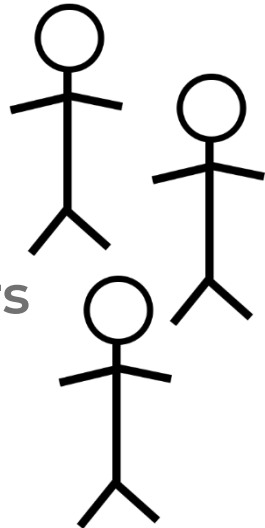
Info-Tech Insight

Swim lane models were a state-of-the-art process modelling method... in 1940. Use case models remain the best model we have to design human-computer interaction, and have continually evolved to support delivery of digital systems using Agile, DevOps, and BizDevOps methods

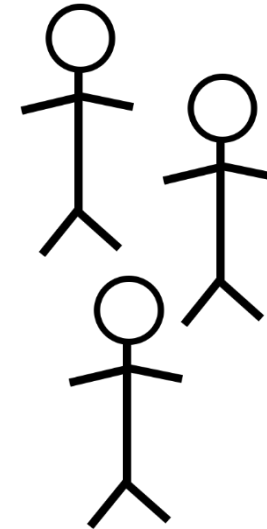
What is UCI for the SDLC?



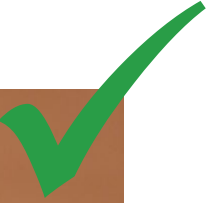
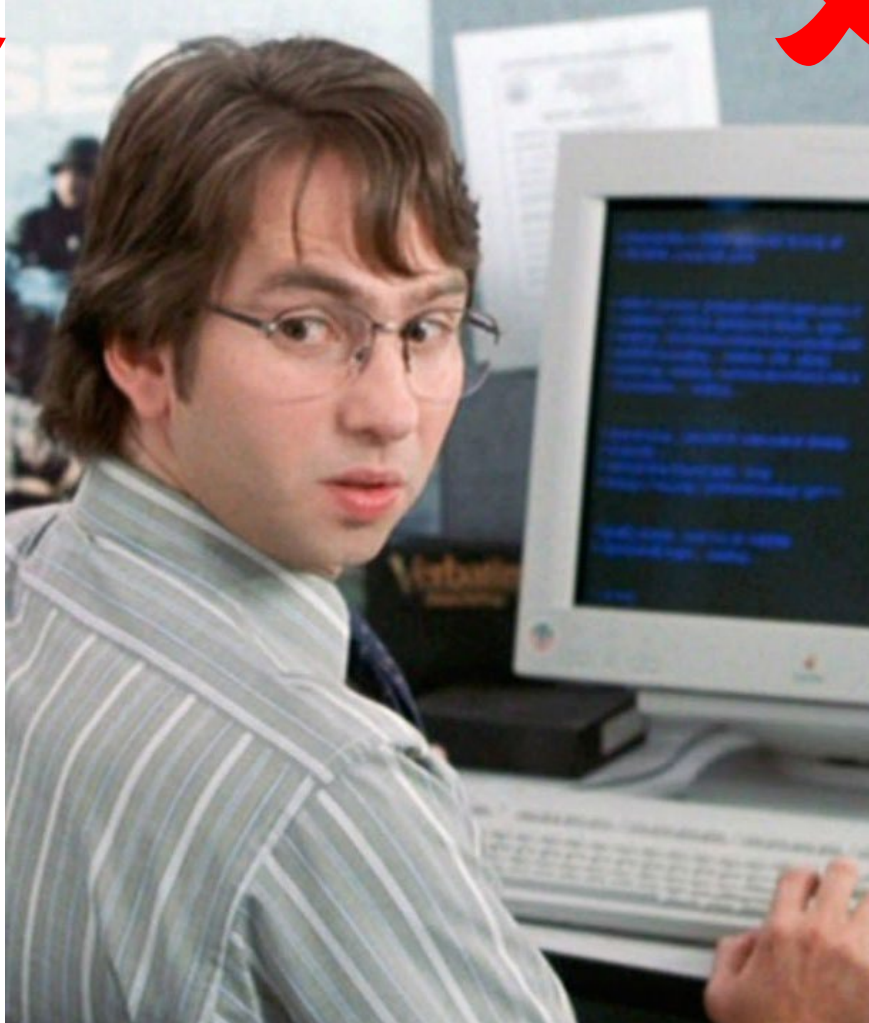
Stakeholders



The Team



Michael Bolton says...

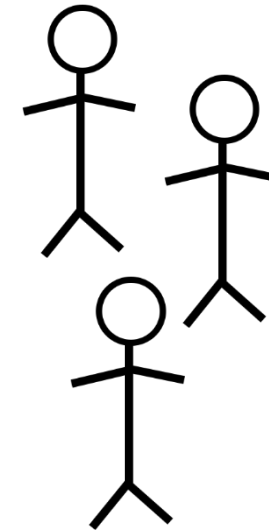
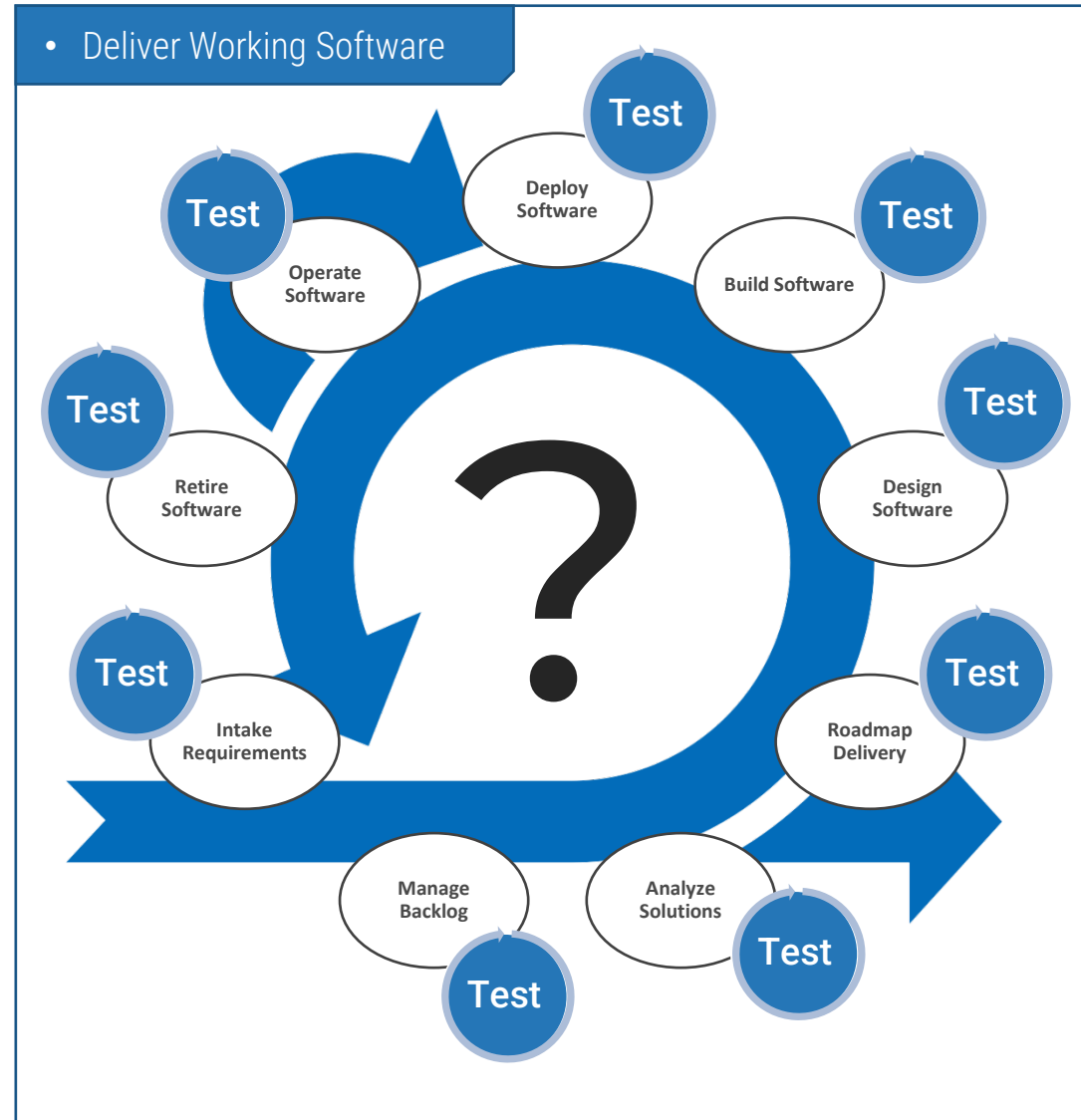


"Testing is... (a lot of things)"

What is UCI for the SDLC?



Stakeholders



The Team

Michael Bolton says...

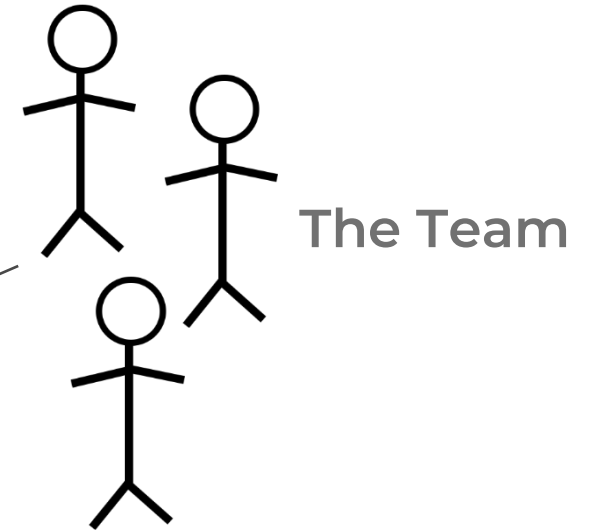
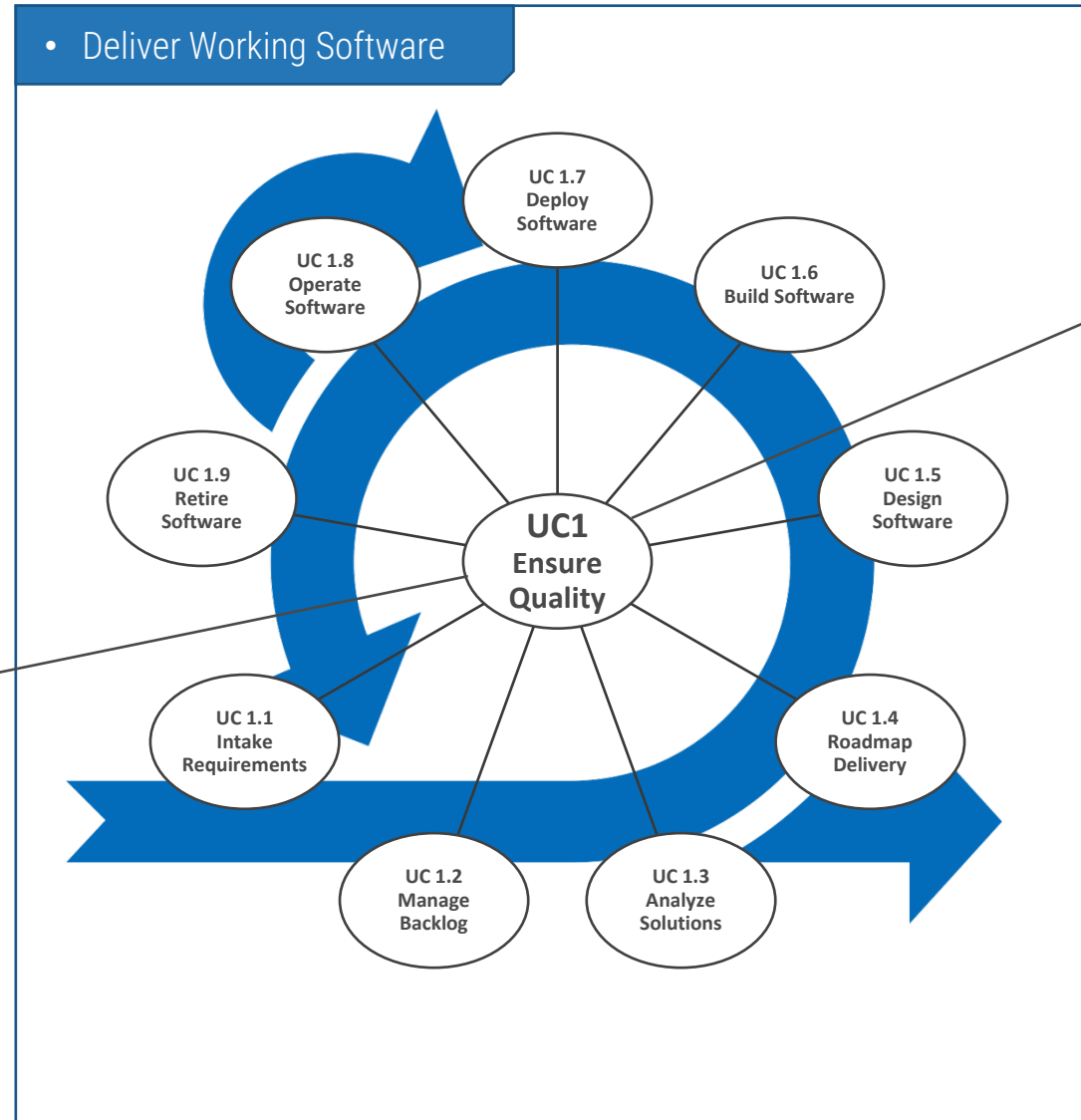


*“Testing is... – among other things –
something that informs quality
assurance, but is not in and of itself
quality assurance.”*

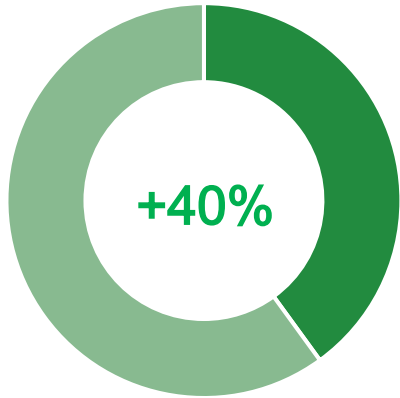


Quality is Use Case 1

That doesn't mean you should implement an automated testing tool first!

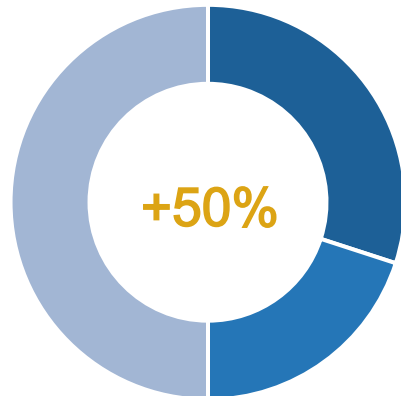


Step 2: Stop writing testing (and other) documents



Increased efficiency of operations of data-centric digital processes over document-centric processes

Source: Forbes, 2019



Customer and User satisfaction increase 50% with data-centric digital processes.

Source: McKinsey, 2019



Digital does not mean no documents. It means no one will ever write one again.

While documents have sections and headings, this does not create sufficient structure to allow automated processes to readily use the content. In the language of data management documents are a “data jail” that traps content in a black box that can only be managed through inefficient, manual processes.



Document-centric

Document-centric means documents first.

Actors create unstructured data in documents that are manually validated and communicated via email (or worse printed out) to execute on the process.

Interactions related to the documents are managed separately, at risk due to multiple drafts or versions, or worst of all lost.



Data-centric

Data-centric means actors create (or even better reuse) structured data that is automatically validated and managed by applications. All interactions are conducted through the application, and documents, when required, are generated by the application from that data and viewed in it as a baseline at a specific point in time.

Your requirements or product backlog items are the building blocks for quality

It doesn't matter whether you are a Waterfall, Agile, DevOps, or somewhere in between!

| Agile | Waterfall | Relationship | Definition |
|----------------|-------------------------|--|---|
| Raw Idea, Idea | | Is realized by one or more... | A valuable yet partially defined goal or objective that requires further analysis from various teams. |
| Epic | Business Requirement | Is realized by one or more... | A statement of a goal or objective that can be estimated and has a defined business value to the organization. |
| Capability | Stakeholder Requirement | Is realized by one or more... | A product or service that one or more stakeholders needs in order to satisfy the business requirement or epic and has a measurable value to the organization. |
| Feature | Functional Requirement | Is constrained by one or more... | Functionality and information the solution needs to provide stakeholders in order to satisfy the stakeholder requirement and has a specified value to the stakeholders. |
| User Story | N/A | "A placeholder for a conversation" – Dr. Alistair Cockburn | User stories are useful tools for managing requirements, design, development, and tests. Don't confuse them with the requirements you need to manage your product backlog and roadmap and deliver a quality solution. |
| Task | Activity | One or more per artifact... | Something the delivery team must do to satisfy requirements. |



**Test
Coverage?**

*“A placeholder for a
conversation”*
– Dr. Alistair Cockburn

Analyst Perspective

Manual Testing isn't Quality



Time spent performing routine cognitive tasks like testing are, at best, a distraction from the more important question of how to assure that we deliver a quality product. Specifically, quality does not just mean the software works as expected,

but that the software creates the expected value.

Cole Cioran

Head of Applications Research and Advisory Services
Info-Tech Research Group

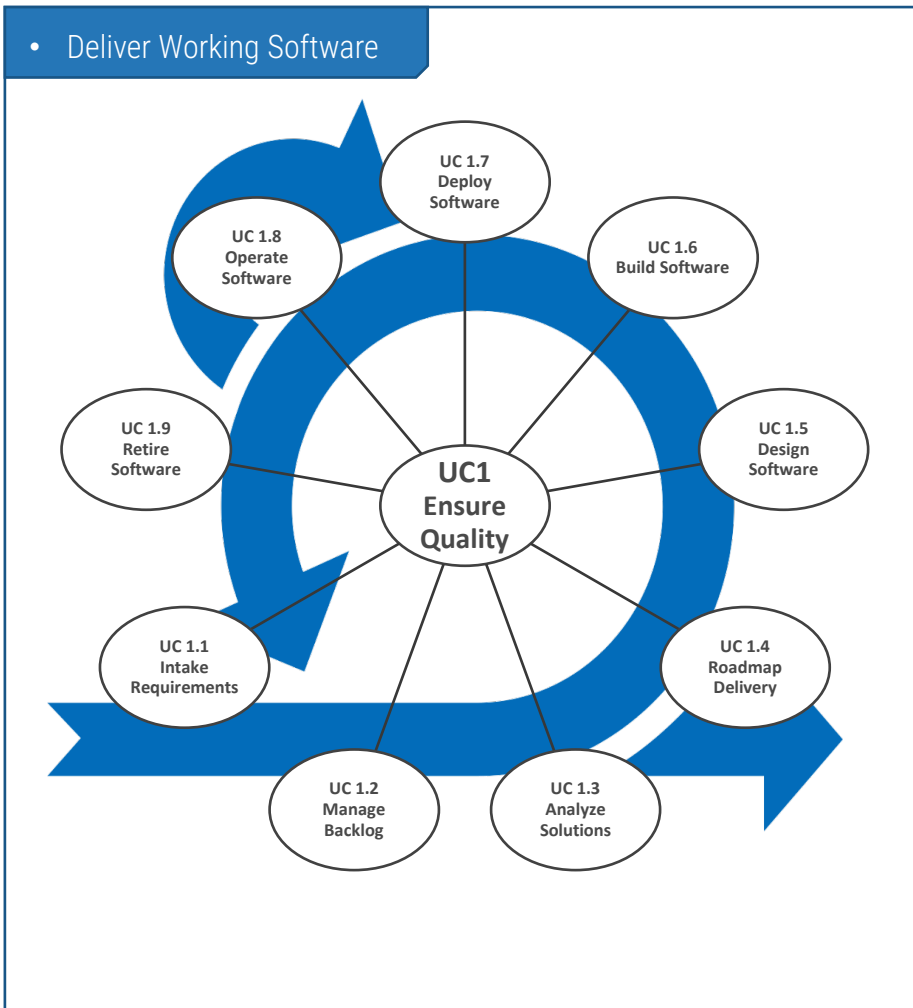
Remember the latest technology?

The Object Model Group's Methods and Tools Platform Specific Interest Group studied 135 SDLC tools and found:

1. Customers treat brands as if they were a religion
2. *Despite* significant overlaps in features and functionality
3. Features are usually captured in the vendor platform
4. The complexity and overlap keeps increasing

- Sumeet Malhotra, OMG Group Chair, Expert Interview, 2021

Step 3: Target optimization and automation where it matters most.



- 1 Start with activities that create the greatest benefit for the most people
- 2 Target the biggest bottlenecks in the Software Delivery Life Cycle
- 3 Integrate the best tools for the job
- 4 Let teams engineer the product AND the product line

“Complete optimization of technical processes is great for instances where the same output is enough to satisfy business needs. However, in cases where sources of business value keep evolving, *optimizing the entire value chain can become difficult and sometimes counterproductive.* In such cases, optimize and automate tasks that have low probability of changing. For the remainder, use what works best. In the end, overall quality of your technical processes should determine your plans.”

– Haroon Shaikh, Engineering Manager, Flipp

3.1 Start with activities that create the greatest benefit for the most people

1 What makes up a viable product vision?



Who is the target customer?



What do they need?



What is the key benefit?



What is the differentiator?

2 What informs the product vision?

Goals

Product goals to achieve your vision.

Metrics

Criteria to determine if we attained our goals.

Value

The value that needs to be provided to meet your goals.

Audience

Who is interested in vision attainment?

Personas

Who is the value being provided for? How is it being provided?

Operational Engagement

3 How do you refine the product vision?

The vision is developed through four key product artifacts:

Product Canvas

Asserts the overall product vision through organizing your thoughts based on the elements of the vision.

Product Backlog

Organization of elements

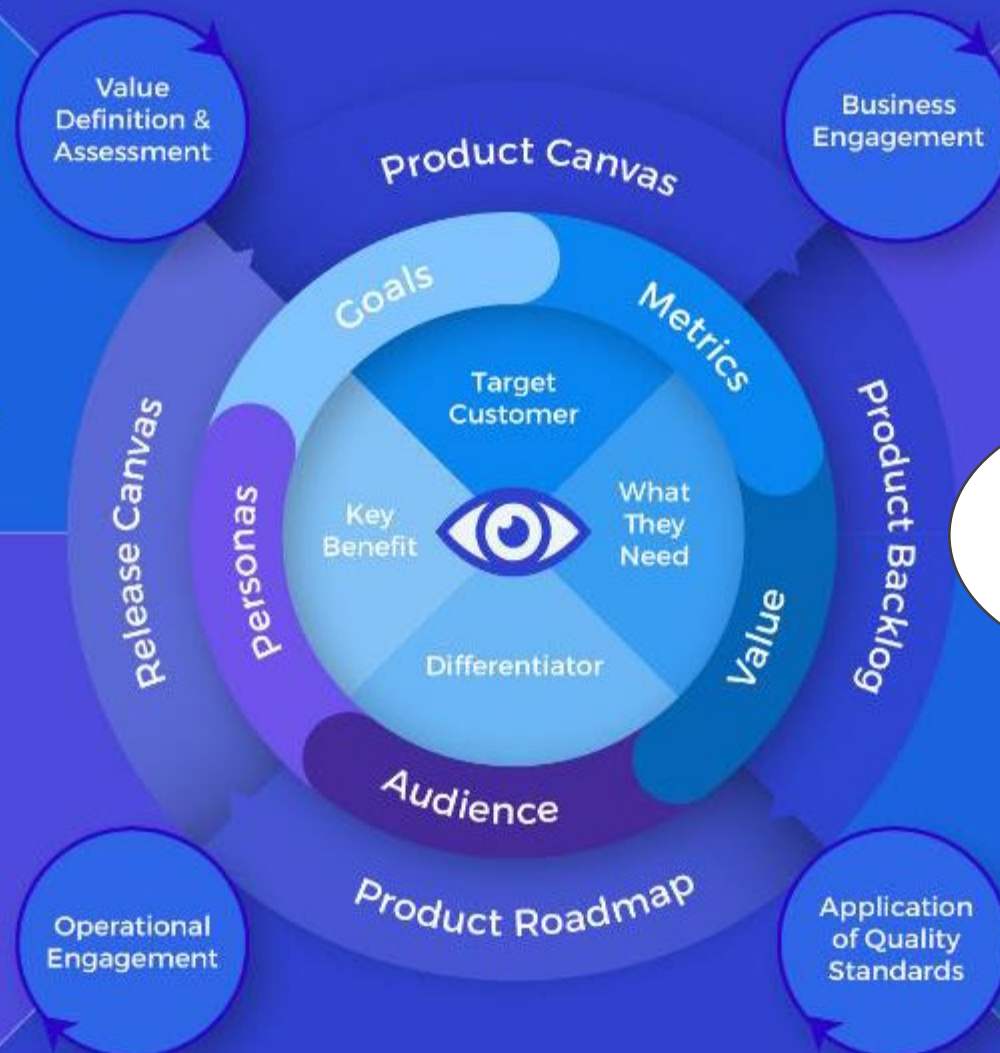
Product Roadmap

Communicates the backlog to stakeholders (internal and external).

Release Canvas

Based on the roadmap and backlog, lays out the elements to drive the next release(s).

UC 1.2 Manage Backlog



UC 1.3 Verify of Product vision!

All enablers are part of an evergreen product management process.

A backlog is enabled via continual and regular **stakeholder engagement**.

Applying **quality standards** is important in helping to refine the roadmap.

Operational engagement is required to lay out a release canvas.

Your release plan combined with **value definition** will confirm alignment to your goals and vision in the product canvas.

3.2 Target the biggest bottlenecks in the Software Delivery Life Cycle

Turn Up Throughput With Automated Testing

INFO~TECH
RESEARCH GROUP

#ITRG



Turn on Foundational Practices

ON MANAGEMENT

Apply and Enforce Quality Standards
Maintain a Testing Artifact Repository
Adopt a Test Management Solution
Actively Monitor and Tune Your Tests

ON TESTING TOOLS

Ensure Your Tools Are Fit-for-Purpose
Architect Your Tools
Routinely Rationalize Your Toolset

ON ORCHESTRATION

Coordinate Automated Testing With Other Delivery Tools
Centrally Orchestrate Automated Testing

ON CONFIGURATION

Modularize and Batch Your Test Runs
Build in Self-Healing and Script With Repeatability in Mind

ON TEST DATA

Apply a Lightweight Data Management Model
Mask and Obfuscate Your Test Data With a Plan
Develop a Synthetic Data Generation Practice

ON TEST ENVIRONMENT

Apply a Lightweight Environment Management Model
Mock and Stub Your Environments With a Plan

ON GOVERNANCE

Define Automated Testing Ownership
Empower and Automate Approvals
Govern Automated Testing Like a Software Product
Build an Automation Community of Practice
Create Transparency for All Stakeholders

ON APPLICATION SECURITY

Test Vulnerabilities to Internal and External Factors
Adopt Static and Dynamic Application Security Testing
Test Incident Response and Application Recovery Workflows
Build Security Policy Enforcement Into Your Tools
Test With Random and Chaotic Data

Start by automating unit and functional tests. They are simple to implement, contained, and based on easy-to-use templates, tooling, and frameworks. Use the lessons learned as building blocks to implement more complicated automated tests.

Mature automated testing for one application first. Build the automation skills, artifacts, processes, and cross-functional relationships that serve as a proof of concept and reusable framework to extend automated testing to other applications.



Novice

Establish your code analysis, unit, and functional tests as bases for all other tests.

Unit Tests – Test the functionality and stability of specific lines of code and configurations in isolation.

Code Analysis – Analyze lines of code against defined coding rules and standards.

Functional Tests – Test a specific action or function of an application against functional requirements and their acceptance criteria.

UC 1.6
Build
Software



Developed

Maintain source code integrity by triggering your component, build, and regression tests alongside your daily code check-ins and builds.

Component Tests – Test individual application components or modules without integrating them with other components or modules.

Component Integration Tests – Test the interfaces and the interactions between integrated components.

Build Tests – Test the build's integrity, stability, and capability to be deployed on desired environments in various scenarios.

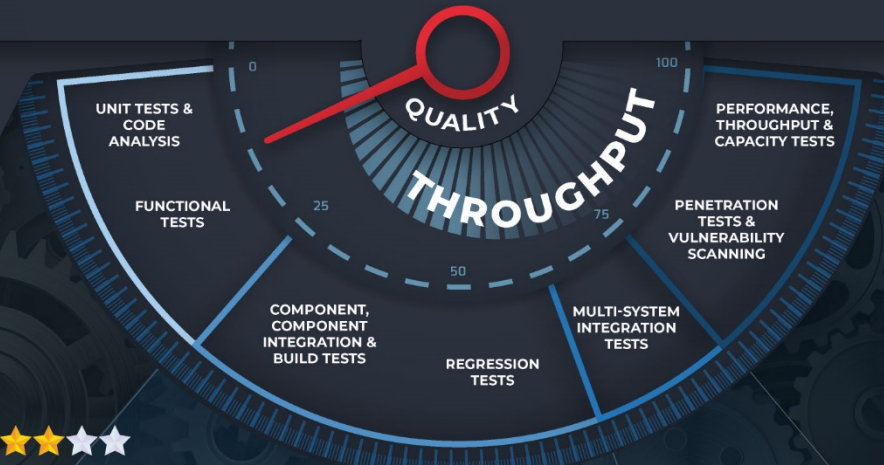
Regression Tests – Test to ensure application changes have not adversely impacted unchanged areas of the application and integrated systems.



Mature

Instill integration tests as a condition of development completeness rather than a prerequisite of production readiness.

Multi-System Integration Tests – Test the application interfaces and the interactions between integrated applications and systems.



Boss

Set system efficiency and resilience as your penetration and performance testing goals – not defects.

Penetration Tests – Test an application's resilience to simulated cyber attacks to identify weaknesses that can be exploited.

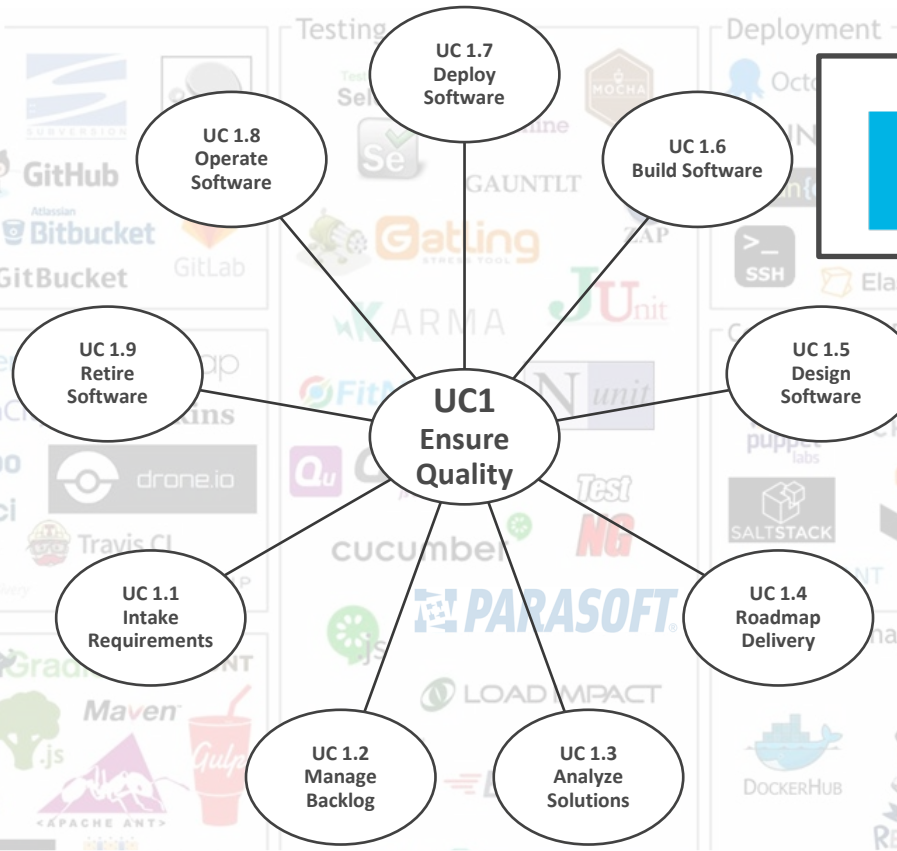
Vulnerability Scanning – Scan the inventory of an application system against a list of known vulnerabilities.

Performance Tests – Test the acceptable time to process a single transaction in isolation or under load.

Throughput Tests – Test the acceptable number of transactions that can be processed in a given timespan.

Capacity Tests – Test the maximum throughput the application can sustain for a given workload while maintaining an acceptable response time for each individual request.

3.3 Integrate the best tools for the job

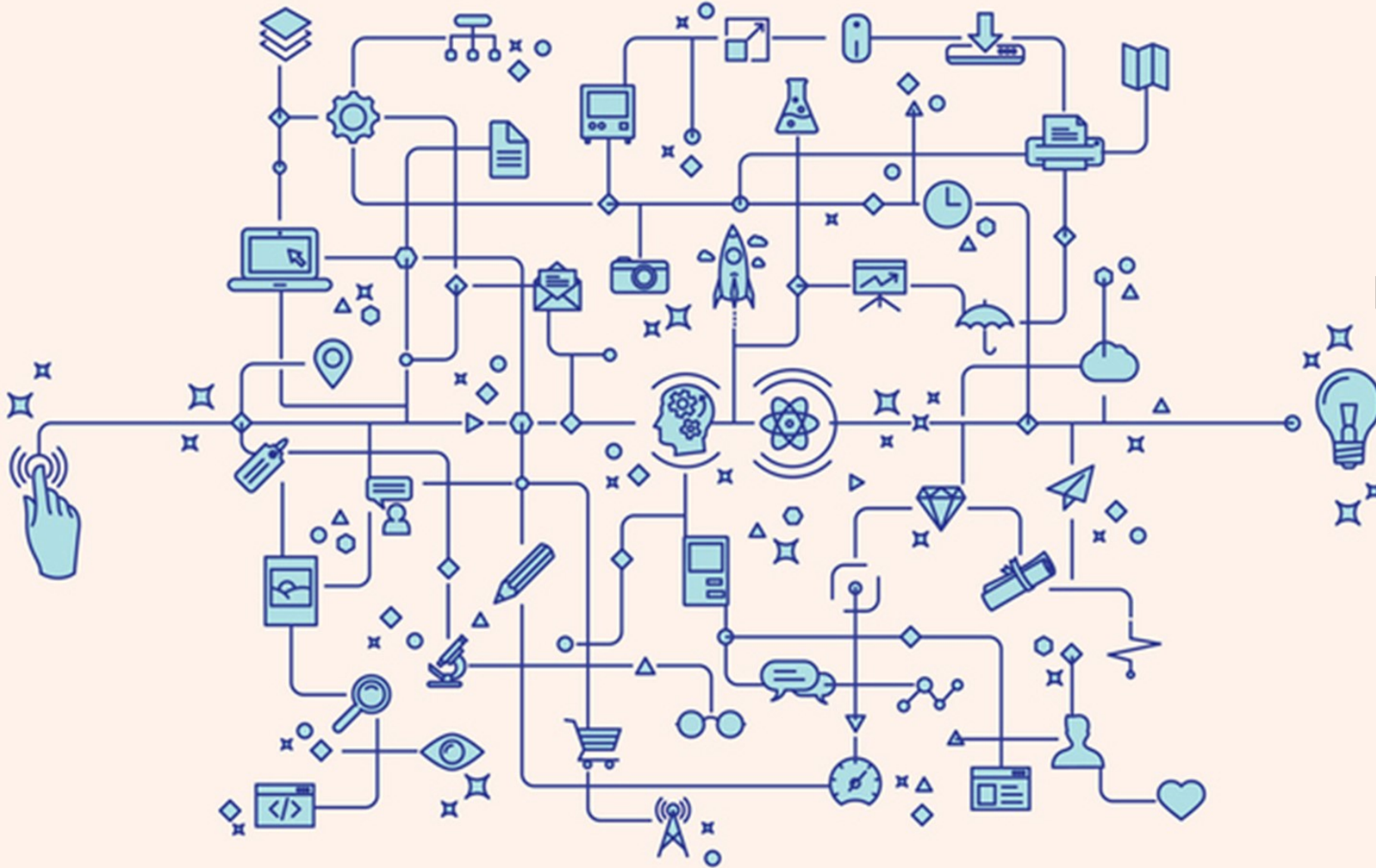


Info-Tech Insight

Choosing leading edge solutions for critical features of strategic capabilities such as software delivery is an important consideration. More importantly, leading-edge solutions integrate, and value stream orchestration solutions integrate your tools and give you the insights you need into the effectiveness and efficiency of your SDLC.

3.4 Let teams engineer the product AND the product line

Before a line of product code is written the software engineering team needs to ensure it has the leanest process enabled by an integrated chain of tools that is required support the build of the product.



An organization's people are its most valuable capital assets, with technology teams being essential in this category.

They understand the business deeply and are de facto drivers of innovation. To ensure their value keeps increasing, training them in the latest tools, technologies, and processes must be a strategic priority.

1

Reimagine your SDLC.

Recognize that automation without optimization reinforces the status quo and is a recipe for long-term pain. Do it right the first time.

2

Stop writing testing (and other) documents.

Documents perpetuate low efficiency legacy SDLC methods and make it hard for software teams and stakeholders to understand whether they are delivering the right thing.

3

Target optimization and automation where it matters most.

Targeted tooling on the least effective parts of the SDLC to reduce bottlenecks and create end to end visibility into the effectiveness and efficiency of the SDLC.

AUTOMATED
SOFTWARE TESTING
AND QUALITY



Hosted by  **PARASOFT**

Thank you for joining!

Manual Testing isn't Quality

Break the vicious cycle of manual software testing and
deliver better software faster!

Presented by Nicholas Cioran
Head of Applications Research and Advisory Services

Info-Tech Research Group Inc. is a global leader in providing IT research and advice. Info-Tech's products and services combine actionable insight and relevant advice with ready-to-use tools and templates that cover the full spectrum of IT concerns.

© 1997-2020 Info-Tech Research Group Inc.

INFO~TECH
RESEARCH GROUP

Who we are

Info-Tech is a global provider of best-practice IT research advisory services that make every IT leader's job easier.

35,000
Members
sharing best
practices you
can Leverage



Millions spent
developing
tools and
templates
annually



Leverage direct
access to over
100 Analysts as
an extension of
your team



Use our
Massive
Data-Base of
Benchmarks
and Vendor
Assessments



Get up to
speed in a
fraction of
the time



Info-Tech Research Group Inc. is a global leader in providing IT research and advice. Info-Tech's products and services combine actionable insight and relevant advice with ready-to-use tools and templates that cover the full spectrum of IT concerns.

© 1997-2020 Info-Tech Research Group Inc.

INFO~TECH
RESEARCH GROUP