No Stone Unturned

The journey of getting from one release a year to Continuous Delivery

@kaimaetzel
kai-uwe_maetzel@us.ibm.com

Lead and Chief Architect of Rational Team Concert
IBM Rational
Overview

- Product and Team
- Goals
- Automation
- Development Process

Many of the aspects have been discussed in the software development methodology, project management, management, behavioral psychology and economics literature.

This is a concrete example of what we do and why.
PRODUCT AND TEAM
Rational Team Concert

- Team collaboration tool combining
  - agile and formal planning
  - task and defect tracking
  - source code management
  - build, packaging, deployment, and promotion
  - dashboards: queries and reports on life and historic data
  - configurable process

- From small teams to large teams of teams

- From MS Windows to z/OS

- Eclipse, MS Visual Studio, Web, command line, MSSCCI, Window Shell integration, ISPF

- Open protocols, numerous integrations

- Hosted and on premise
Team

- Portland
- Ottawa
- Boston
- Raleigh
- Nantes
- Zurich
- Beijing
- Tokyo
- Bangalore
- Perth
- Costa Mesa
- Toronto
- Paris
- Bangalore
- Perth

© 2013 IBM Corporation
RTC Trivia

- We develop RTC using RTC
- We develop transparently on jazz.net and run JazzHub
- We are using a mix of Scrum, XP, Kanban, … evolving since 1999 originally known as the Eclipse Way
  - strong skill set
  - Hundreds of builds a week (person, team, team of teams, project)
  - measure “everything”
  - …
Observations

- Annual release cycle causes unbearable pain
  - up to two years to delivery of customer requests
  - endless renegotiation of release content
    - unfinished worked, broken flows
    - wasted efforts
    - worn down development and test teams since the last agreed upon release content is late

- Customers have a hard time understanding what comes and what doesn’t

- Milestone builds received feedback but not to the amount we need: too much work for customers for non-production use
What do we do?

- Release more often, we settled on quarterly
- Use processes that optimize decision making, development workflows, and transparency
- Only change direction at well defined breaking points
- Automate as much as possible
- Shorten time to self-hosting and deployment in cloud offerings

Agile

Continuous Delivery
Game Plan

- Develop the necessary build, deploy, and test infrastructure
- Fill the gaps in automated testing
- Rework development and delivery process
BUILD, DEPLOY, TEST
Automation

- “If you need a machine and don't buy it, you will ultimately find that you have paid for it and don't have it.” (supposedly Henry Ford around 1920)
- Automation work is represented in our backlog. We don’t treat it special.
- We heavily weight the impact on operational effectiveness during the backlog prioritization process

Automation Milestones

1. Every build can automatically be verified, deployed, and setup
2. We auto-deploy to the jazz.net staging environment
3. We enable ourselves to auto-deploy builds to jazz.net
4. We enable ourselves to auto-deploy builds to JazzHub
Deployment pipeline takes the output of a build, automatically deploys and runs it through a series of progressively more rigorous, time-consuming tests.

Enables testers and developers to trivially provision a new environment for manual testing or debugging.
CLM Build Example

- BVT child build
- BVTCO child build
- BVTPRT child build
- BVTQM child build
- BVTSVT child build
- CLM child build
- RIF child build (reused RIF-i20130321-1931)
- RQM child build (reused RQMMainline-i20130322_0448)
- RRC child build
- RTC child build (reused RTC-i20130322-0211)
- RTCDEPLOY child build
CLM Build Example

![Build jbslnxvh08.deploy.40 20130324-1944](image)

**Completed**
- Duration: 33 minutes, 17 seconds
- Start Time: March 24, 2013 5:08:58 PM
- Completed: March 24, 2013 5:42:16 PM
- Status Trend: [progress bar]

**Contribution Summary**
- Changes: [Show changes]
- External Links: [1 link]
- Logs: [3 logs]
- Repository Workspace: [RTC Integration 3.0 Server Deploy Workspace]
- Snapshot: [jbslnxvh08.deploy.40_20130324-1944]
- Work items: [1 included in build]

**Associated Release**
- Released builds are available as choices in the work item "Found in" field.
- [Create a release to associate with this build]

**General Information**
- Requested by: Mr. Build
- Build Definition: [jbslnxvh08.deploy.40]
- Build Engine: [jbslnxvh08]
- Build History: [40 builds]
- Tags: [calm.all=20130324-1457]
- Deletion allowed: [✓]
CLM Build Example
Start Small

- Since we could not get there in a single step we built simpler deploy scripts and provide context menu actions on build results in eclipse to deploy the builds to dedicated test servers in the cloud.
Testing

- Before we could survive when we had test coverage independently of whether the builds were in a good shape. We provide guidance to all teams and rework all tests.

- We continuously drive up test coverage.

- There are no excuses for new functionality.

- Reducing debt is prioritized through the backlog grooming process.

Unit Testing In The Real World

This article is an opinionated guide on how to unit test complex software systems. It differentiates Mockito as a mocking tool and shows how it can be used to create maintainable mock the genuine program behavior, such that created mock tests will contribute to improved wrapping a safety net around your program, it is important to know which of its parts is covered workspace and in continuous builds to identify untested program parts.

Unit Testing vs. Integration Testing
When To Write Unit Tests
Isolate System Behavior With Test Dummies
How To Mock, Stub, And Spy
Creating Mocks With Default Answers
Define Stubbed Values For Methods
Under-specify The Mocking With Matchers
Inject State Into Mock Objects
Spies Are The Better Mocks
Verification Of Indirect Output
Beware Of The Mock
Test Coverage
Test Coverage In Your Workspace
Test Coverage In Continuous Builds
References
DEVELOPMENT PROCESS
Setting The Stage

- Culture is the unique combination of processes and priorities. Culture is formed by repetition. “That’s the way we do things.”

- You need to internalize that you are a drunk before you can get better.

- You need to understand the theory of marginal and absolute costs.
  - Remember Henry Ford.
Development Process

- Ship installable products every 12 weeks
  - TVT, SVT, Qcert, installer production, legal, publishing

- Ship customer value every 12 weeks
  - when a feature is declared done, it’s shippable and can be supported in the field

- Optimize
  - team effectiveness (market impact, productivity)
  - product quality
  - stakeholder/customer transparency
## How does the Process help?

<table>
<thead>
<tr>
<th></th>
<th>Ship It</th>
<th>Ship Value</th>
<th>Revenue</th>
<th>Productivity</th>
<th>Quality</th>
<th>Stakeholders</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clear core values</td>
<td>✭</td>
<td>✭</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Releases each quarter</td>
<td>✭</td>
<td></td>
<td>✭</td>
<td>✭</td>
<td>✭</td>
<td></td>
</tr>
<tr>
<td>Run teams</td>
<td>✭</td>
<td></td>
<td></td>
<td>✭</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Automation of build, test, and deployment</td>
<td>✭</td>
<td>✭</td>
<td></td>
<td>✭</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Continuously maintained roadmap</td>
<td>✭</td>
<td>✭</td>
<td>✭</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prioritized product backlog</td>
<td>✭</td>
<td>✭</td>
<td>✭</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Single reference &amp; TOC of deliverables</td>
<td>✭</td>
<td></td>
<td></td>
<td>✭</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Start release cycle with release blog</td>
<td>✭</td>
<td></td>
<td></td>
<td>✭</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Comprehensive Definition of Done</td>
<td>✭</td>
<td></td>
<td></td>
<td>✭</td>
<td></td>
<td></td>
</tr>
<tr>
<td>One Feature Team per plan item</td>
<td>✭</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Free-floating senior technical people</td>
<td>✭</td>
<td>✭</td>
<td></td>
<td>✭</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monitor diligently</td>
<td>✭</td>
<td>✭</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Communicate process clearly</td>
<td>✭</td>
<td>✭</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

More information: [https://jazz.net/wiki/bin/view/Main/RTCContinuousDeliveryPlanning](https://jazz.net/wiki/bin/view/Main/RTCContinuousDeliveryPlanning)
Communicate Process Clearly

Rational Team Concert Dashboard

What You Need to Know
- Learn about our Way of Working
- Have a look at our Product Backlog
- Have a look at our Roadmap for the next 6-18 months
- Learn more about our next release

RTC Product Backlog (CCM)

215 items: 196 open, 19 closed | Ends in: --

Plan Details

Guidance

Our planning process is described at https://jazz.net/wiki/bin/view/Main/RTCContinuousDeliveryPlanning.

The structure of our backlog is explained at https://jazz.net/wiki/bin/view/Main/RTCContinuousDeliveryPlanning#ProductBacklog.
Communicate Process Clearly

Plan Item 213872

Summary: [CCM] Improve web UI performance for large plans

Overview

Type: Plan Item
Project Area: Rational Team Concert
Team Area: PMC
Filed Against: PMC
Theme: Quality and Performance

Description

Information about plan items and our planning process can be found at https://jazz.net/wiki/bin/view/Main/RTCPanItemCheckListExplained
Communicate Process Clearly

All artifacts are linked for easy navigation
Core Values – Independent Decisions

- Seamless task flows.
- Best of breed.
- Your way.
- Deep analysis.
- Team focused.
- Across teams, functions, projects.
- Multi-platform.
- Extensible.
- Scalable.
- Flexible deployment.
- Secure.
- Safe investment because of open resource formats and stable APIs.
SHIP IT
Run Team

**Rules of the Road**

**Inbox Tracking (Inboxmeister):**
- Track the work items filed against Planning and Workitems components; those are not assigned and triaged - TAP Inbox
- Update the severity of the work items as per Bug Severity
- Update the work item with appropriate tags and category; TAP Triage Tags
- Investigate or assign the work item
  - for investigation add the Investigate tag and assign to a run team member
  - to fix an item, assign it to a run team member and set the target to the next milestone
- After investigation is done, remove the tag and
  - Assign it to the team to fix and set to the next milestone
  - Move to the Backlog

**Forum Tracking (Newsmeister):**
- Monitor the forum at jazz.net for the questions related to Planning and Workitems component - Unanswered Forum Questions
- Get a timely response to their questions
  - Answer the question yourself
  - Reach out to someone in the team
- Tag items that need a response that you don’t have time to make right away with “tap-help” - TAP-Help Questions
- Remove the tag if it is answered

**Build Meister Backup:**
- As a backup, share the load of the buildmeister as per circumstances
- Follow up on the issues related to the builds
- Update the status of the builds in weekly component lead call

**APAR/RFE Tracking:**
- Review APAR/RFE candidates. Decide if the request is a valid APAR of a RFE: Pending Reviews
- Plan and Fix APARs
- Review Backports
The “Theory of Run Teams”

- Responsibilities
  - interact with the community
  - manage the inbox
  - prepare and execute build submissions
  - pay for the sins of the past

- Run teams are led by Run Team leads. Run Team leads offload component leads.

- Number of defect fixes necessary indicates how well or poorly we executed in the past.

- If we execute well, the load on the run team will go down in the future. (Normalization by adoption rate may make the absolute numbers look differently.)

- You need to manage behavior not quality gates. Gates are good as KPIs. If you manage towards gates teams will achieve the numbers but not quality.
Permanently Tracking Quality Goals
Guidance What to Focus On

Defect Count by Age

Ship it
Streamline all “Must Haves”
- Tales of “It really takes so long!”

- TVT
- SVT
- installer production
- legal reviews
- publishing
SHIP CUSTOMER VALUE
Prioritized Product Backlog

- Plan Item F
- Plan Item C
- Plan Item A
- Plan Item D
- Plan Item B
- Plan Item E

- Theme 1
  - Backlog Item 1.1
  - Backlog Item 1.2
  - Backlog Item 1.3
  - Backlog Item 1.4

- Theme 2
  - Backlog Item 2.1
  - Backlog Item 2.2

- Theme 3
  - Backlog Item 3.1
  - Backlog Item 3.2
  - Backlog Item 3.3
  - Backlog Item 3.4

- Backlog Item 4
- Backlog Item 5
- Backlog Item 6
- Backlog Item 9
- Backlog Item 11
- Backlog Item 34

Clarity

Emerging Understanding

No Understanding
+ all items assigned to “Backlog”
What are the Items on the Prioritized Product Backlog?

- No more than 8 weeks of work when in the clarity section.
- One concrete issue or a theme of related issues.
- Broken down into stories for execution
- We call them Plan Items.

- Items in the Clarity section are ready to be worked on.
- Plan item moves to the release plan only when staffed and execution starts.
- Enhancement Requests make it to the PPB through their linked plan item.
- If not linked, they are in the “No Understanding” section.
Single Access Point and TOC of Deliverables

26. new and noteworthy
   - deliverable
   - Prepare new and noteworthy (220044)

27. demo video
   - Video - Part 1
   - Video - Part 2
   - Prepare demo video (220045)

28. additional support enablement materials
   - Table of content of all materials to support feature (220046)
   - Script-based decimal example
   - Decimal Support Slides

29. explanation of how to support the feature in the field, document available tracing/debug interface info, ide
   - Create supporting jazz.net material (220338)
Continuously Maintained Roadmap

Rational Team Concert Dashboard

What You Need to Know
Learn about our Way of Working
Have a look at our Product Backlog
Have a look at our Roadmap for the next 6-18 months
Learn more about our next release

RTC Product Backlog (CCM)

215 items: 196 open, 19 closed | Ends in: --

Plan Details

Planned Items Links Snapshots Dashboard Notes Roadmap

Proposed IBM Rational Team Concert Roadmap 2012-2013:

We have a vision for IBM Rational Team Concert to be the best team based collaborative development solution. Our view of managing on time delivery of software. But we also offer a key value to external stakeholders. These are all the people engaged in a project that they depend on so they can prepare the launch of a product, or the CIO who needs to explain to the board its focusing on usability and new software delivery models that enable teams to get started with IBM Rational Team Concert with...
Release starts with Release Blog

**RTC 4.0.1 Release Plan**

34 items: 2 open, 32 closed | Ends in: 17 days

Plan Details

**Rational Team Concert 4.0.1**

Like all other x.0.1 releases our 4.0.1 release is a maintenance release. But it is not only that. It also is the first release that was shaped by our tr

At the core of our transformation is the reinvigoration of agile principles:

- **Release often**: We work in 8 week release cycles. Each cycle has two 4 week sprints.
- **Work on business priorities**: We work out of a prioritized product backlog.
- **Right-sized work**: Each item on the backlog fits into one 8 week cycle.
- **Done/Done**: Each plan item has a checklist that we work through in order to complete the item.
- **Stay focused**: A cross-functional feature team works on one plan item from start to finish.
- **No false promises**: When we start working on a plan item it moves onto the release plan.

For you in the community the changes visible so far are subtler but you will notice more and more over time. The change to the release plan alre backlog tells you what will come next.
Definition of Done

Done Criteria:
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- n/a
- n/a
- Pending
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- n/a

Acceptance Test:
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- n/a

UX Design:
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- n/a

No Broken Flow:
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- n/a

Implementation:
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- n/a

Unit Tests:
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- n/a

API Compatibility Tests:
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a

No Broken Platform:
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a

Trace & Debug Support:
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a

Security Review:
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a

Messages Review:
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a

No Broken API:
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a

No New Technical Debt:
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a

Manual Test Cases:
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a

Performance Tests:
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a

Migration Tests:
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a

N-1 Compatibility Tests:
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- n/a

VSIT:
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a

GVT:
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- n/a

TVT:
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- n/a

SVT updated:
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a

FVT updated:
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a

COO:
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a

API Support Statement:
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a

Documentation:
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- n/a

New & Noteworthy:
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- n/a

Demo Video:
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- n/a

API Usage Sample:
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a
- n/a

Support Enablement:
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- n/a

Field Support Guide:
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- Completed
- n/a
Feature Team

- Each plan item is executed by a cross-functional feature team.
- The feature team is lead by a feature team lead (project manager).
- Each feature team is supported by a senior technical and subject matter expert.
- Senior technical team members may form Design Boards.
The "Theory of Feature Teams"

- Each feature team **should be** composed out of Mini Teams. Mini Teams are co-located and undividable. This ensures no person works alone at any site or in any time zone.

- Feature teams dissolve after the plan item is completed. Thus, we ensure knowledge transfer in all directions. Over time feature teams will become more stable.

- Feature team leads offload the component leads. Component leads focus on technical work and act as the senior technical and subject matter experts. Component leads are the most precious resources we have.

- Feature team lead is not a a permanent position or specialization. Everyone needs to be able to play this role to ensure that our values and processes are deeply engrained in every single team member.
Tracking the Completion

| ID  | DC | AC | UI Design | No Broken | Code | Unit Tests | API Tests | No Broken | Test Security | Regression | No Defect | Manual Testing | Component Test | Migration | N+1 | VS | GVT | TVT | SVT | FVT | COD | API | SS | DOC | N & N | Demo | Video | API | Enable | Support | Field | Support |
|-----|----|----|-----------|-----------|------|------------|-----------|-----------|---------------|------------|-----------|--------------|----------------|-----------|------|----|----|----|----|-----|-----|-----|-----|-----|-----|------|------|------|-------|--------|--------|---------|----------|
| 213835 |     |    |           |           |      |            |           |           |               |            |           |              |                |           |      |    |   |   |   |   |     |     |     |     |     |     |     |     |       |         |         |         |         |
| 213872 |     |    |           |           |      |            |           |           |               |            |           |              |                |           |      |    |   |   |   |   |     |     |     |     |     |     |     |     |       |         |         |         |         |         |
| 211105 |     |    |           |           |      |            |           |           |               |            |           |              |                |           |      |    |   |   |   |   |     |     |     |     |     |     |     |     |       |         |         |         |         |         |
| 215178 |     |    |           |           |      |            |           |           |               |            |           |              |                |           |      |    |   |   |   |   |     |     |     |     |     |     |     |     |       |         |         |         |         |         |
| 218178 |     |    |           |           |      |            |           |           |               |            |           |              |                |           |      |    |   |   |   |   |     |     |     |     |     |     |     |     |       |         |         |         |         |         |
| 216545 |     |    |           |           |      |            |           |           |               |            |           |              |                |           |      |    |   |   |   |   |     |     |     |     |     |     |     |     |       |         |         |         |         |         |
| 219545 |     |    |           |           |      |            |           |           |               |            |           |              |                |           |      |    |   |   |   |   |     |     |     |     |     |     |     |     |       |         |         |         |         |         |
| 216548 |     |    |           |           |      |            |           |           |               |            |           |              |                |           |      |    |   |   |   |   |     |     |     |     |     |     |     |     |       |         |         |         |         |         |
| 215558 |     |    |           |           |      |            |           |           |               |            |           |              |                |           |      |    |   |   |   |   |     |     |     |     |     |     |     |     |       |         |         |         |         |         |
| 219734 |     |    |           |           |      |            |           |           |               |            |           |              |                |           |      |    |   |   |   |   |     |     |     |     |     |     |     |     |       |         |         |         |         |         |
| 219748 |     |    |           |           |      |            |           |           |               |            |           |              |                |           |      |    |   |   |   |   |     |     |     |     |     |     |     |     |       |         |         |         |         |         |
| 221780 |     |    |           |           |      |            |           |           |               |            |           |              |                |           |      |    |   |   |   |   |     |     |     |     |     |     |     |     |       |         |         |         |         |         |
| 221775 |     |    |           |           |      |            |           |           |               |            |           |              |                |           |      |    |   |   |   |   |     |     |     |     |     |     |     |     |       |         |         |         |         |         |
| 221958 |     |    |           |           |      |            |           |           |               |            |           |              |                |           |      |    |   |   |   |   |     |     |     |     |     |     |     |     |       |         |         |         |         |         |
| 220999 |     |    |           |           |      |            |           |           |               |            |           |              |                |           |      |    |   |   |   |   |     |     |     |     |     |     |     |     |       |         |         |         |         |         |

Ship value
Track Costs for Future Estimates

![Costs and Estimates for 4.0.1 Plan Items](image)

<table>
<thead>
<tr>
<th>ID</th>
<th>Cost</th>
<th>Unknown Cost</th>
<th>Future Cost</th>
<th>Unknown Future Cost</th>
<th>Total Cost</th>
<th>Total Unknown Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>213896</td>
<td>0</td>
<td>10</td>
<td>16</td>
<td>9</td>
<td>0</td>
<td>19</td>
</tr>
<tr>
<td>213872</td>
<td>139</td>
<td>12</td>
<td>72</td>
<td>4</td>
<td>211</td>
<td>16</td>
</tr>
<tr>
<td>218167</td>
<td>96</td>
<td>54</td>
<td>0</td>
<td>1</td>
<td>96</td>
<td>55</td>
</tr>
<tr>
<td>218173</td>
<td>91</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>91</td>
<td>8</td>
</tr>
<tr>
<td>218176</td>
<td>387.25</td>
<td>71</td>
<td>6</td>
<td>2</td>
<td>393.25</td>
<td>73</td>
</tr>
<tr>
<td>218178</td>
<td>94</td>
<td>36</td>
<td>0</td>
<td>0</td>
<td>94</td>
<td>36</td>
</tr>
<tr>
<td>218540</td>
<td>88</td>
<td>17</td>
<td>8</td>
<td>6</td>
<td>96</td>
<td>23</td>
</tr>
<tr>
<td>218543</td>
<td>18</td>
<td>64</td>
<td>0</td>
<td>0</td>
<td>18</td>
<td>64</td>
</tr>
<tr>
<td>218546</td>
<td>1.5</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>1.5</td>
<td>15</td>
</tr>
<tr>
<td>218582</td>
<td>0</td>
<td>14</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>219734</td>
<td>54</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>54</td>
<td>8</td>
</tr>
<tr>
<td>219740</td>
<td>55</td>
<td>11</td>
<td>0</td>
<td>0</td>
<td>55</td>
<td>11</td>
</tr>
<tr>
<td>221178</td>
<td>0</td>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>221996</td>
<td>35</td>
<td>30</td>
<td>0</td>
<td>0</td>
<td>35</td>
<td>30</td>
</tr>
<tr>
<td>222001</td>
<td>40</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>40</td>
<td>2</td>
</tr>
<tr>
<td>222002</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>229208</td>
<td>108</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>108</td>
<td>2</td>
</tr>
</tbody>
</table>
The Waltz of Run and Feature Teams

- Easy to think “Run Team == Slaving and Feature Team == Shining” **BUT**

- Ultimately Run Team and Feature Team members need the same skills
  - domain expertise, conceptual thinking, test automation, architectural understanding, coding standards and rules, …
  - overlooked defects in features become the run teams problem down the road

- Rotational system
  - You are on the run team for only 1 release cycle.
  - A dedicated representative of each feature team is on the run team in the next cycle.
Monitoring

- Measuring separates Reality from Wishful Thinking.
- We continuously extend and recalibrate what we measure to ensure the process is not only an empty promise.
Engaging Teams and Stakeholders

- Leave as much “how” as possible to the teams. Ensure the teams know that it is their responsibility to fill the blanks. Otherwise they won’t feel empowerment but mismanagement.

- Harvest experience and generate cook books not prescriptions. Use the cook books to accelerate the take off a new run team or feature team
  - dashboard layouts
  - feature team plan template
  - list of contacts
  - ...

- Establish official feedback channel
  - RTC > Stakeholder Feedback
  - CLM > Stakeholder Feedback
THANK YOU