Quality is a Team Sport

Davyd Norris
Senior IT Architect – IBM Rational
dnorris@au1.ibm.com

Agile Method Requires Discipline

With Agile implementation comes a new way of working that requires disciplined collaboration.

In order for teams to be effective and efficient in collaborating and communicating using an Agile development methodology, a variety of approaches are used.

Agile Collaboration Essentials for Quality Management

- Establish good relationships

Teams are built from people from many IBM products and projects:

- Learn who your product's team members are
- Establish accessible repositories for all team members to use
- Learn what other products your product will integrate with and who to contact on each team

Learn who your product's team members are

- Create email /distribution lists
- Know where team members are located and
- be sensitive to time-zone differences
- Create accessible repositories for change management and test tracking for all team members to use- our team uses Rational Team Concert (RTC) and Rational Quality Manager (RQM).

Agile Collaboration Essentials for Quality Management

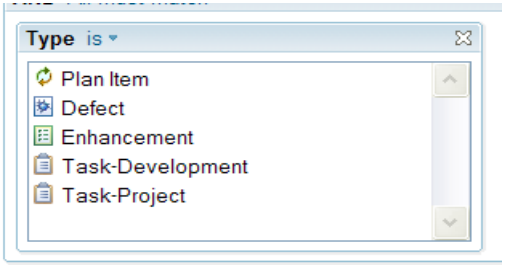
- Participation is key to staying informed

  - Project level meetings (managers and team leads)
  - Iteration meetings
  - Scrum meetings
  - Work product reviews (Test Plans, Test Cases, Product Assessment Reports)
  - Customer meetings and Beta programs
  - Cross-functional team meetings
  - Lessons learned session

How do Rational Team Concert (RTC) and Rational Quality Manager (RQM) assist Agile?

Looking for iteration deliverables information?

- Queries and reports can be created to list iteration deliverables for both products.
  - RQM provides test asset organization, control and reporting features with many product integrations.
  - RTC provides traceability through work item assignments, ownership of work items, identifying related work items as well as parent and child work items.
  - RTC tracks several types of work items:



Collaboration 101

- Establish relationships
  - New definition of “the team”: Developers, QE, UA (Doc), UX, Graphics, Install, Build
  - Iteration deliverables meeting
  - Cross functional team meetings (CFT)
- Drive focus on customer business solutions
  - Use cases from requirements drive development feature deliverables
  - Customer based scenario test cases
  - Role based scenario implementation of test cases (C/ALM)
  - Test gap analysis of customer APARs to improve test coverage in scenarios
- Address Quality
  - Issue tracking and closure
  - Work item subscription, email notification
  - Requirements traceability to test cases (ReqPro)
  - The sooner we can test it, the sooner we can provide feedback to developers

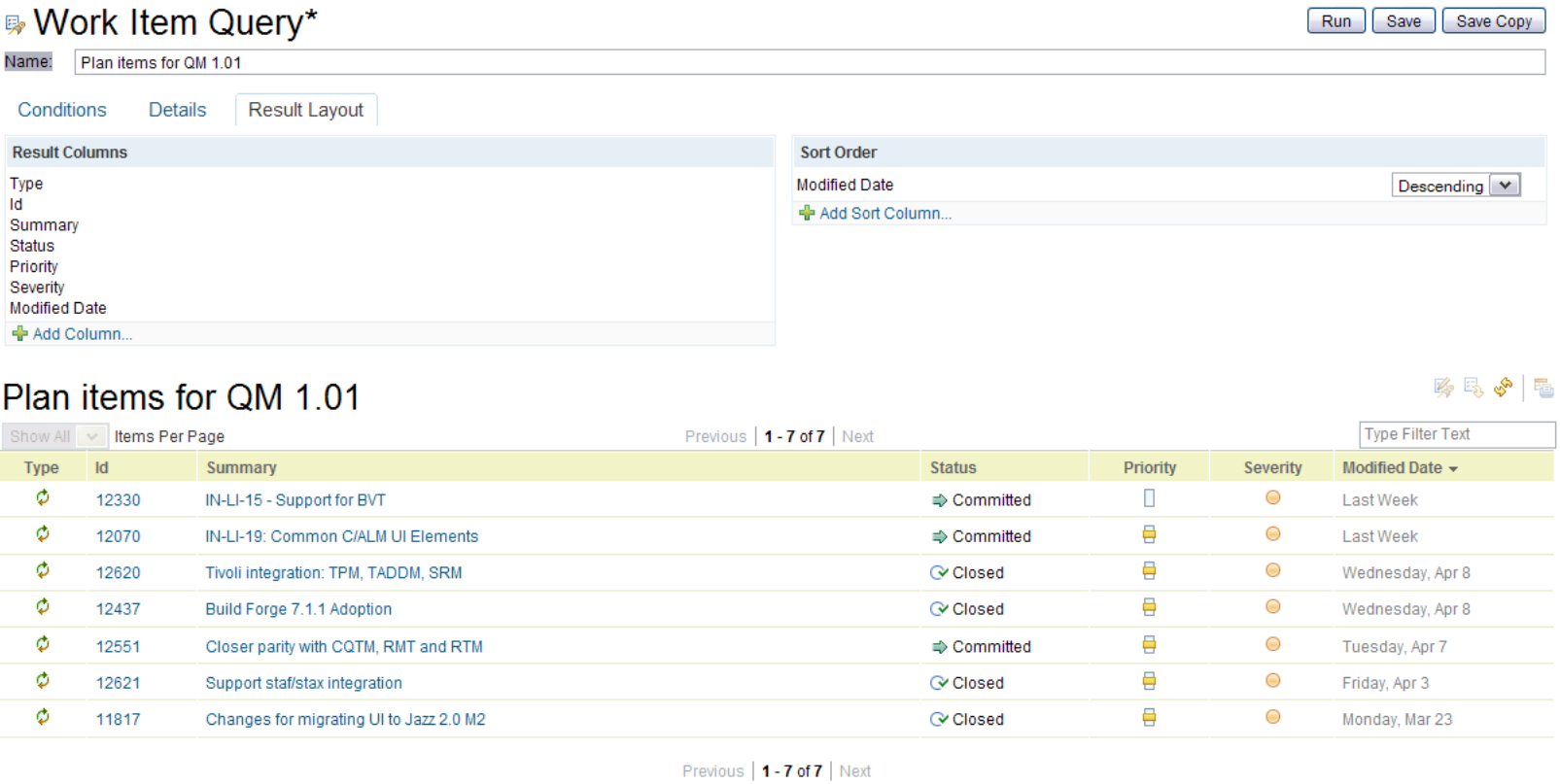
The Team's Agile

- Change in how everyone does things
  - A customer team defines the requirements for each iteration
  - Developers estimate in a massive iteration planning meeting and volunteer for tasks they wish to do
  - 2 week iterations for testing delivered features
    picking up new builds when major defects are fixed to validate
  - Iteration end date does NOT move
  - All features seen in the user interface (UI) work all the way though to the back end datastore

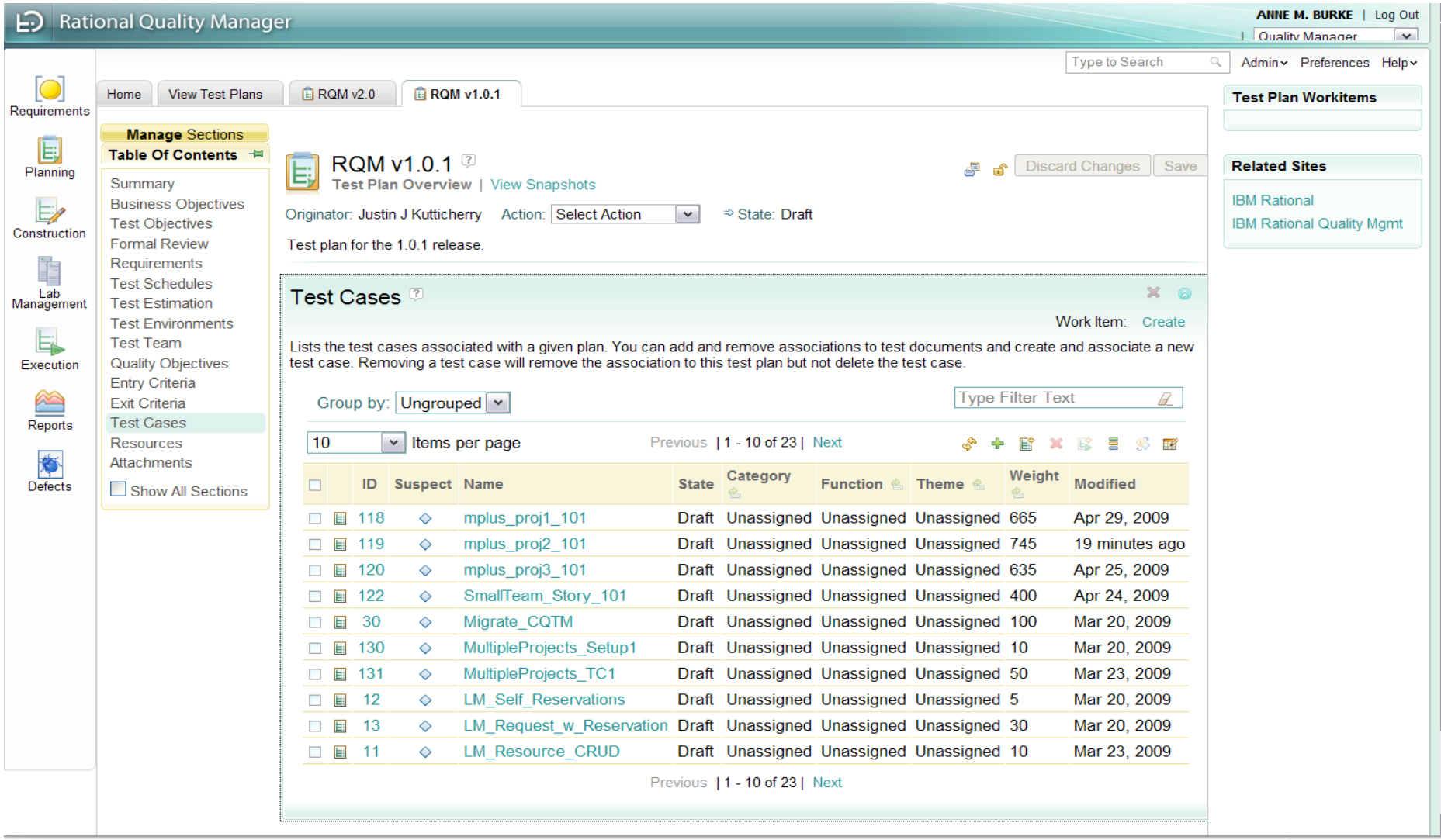
Moments of fear and doubt

- 2 week iterations!! No one can do that
- What kind of tests can we run and when?
- Fine, I will do what I am told
- This will never work! We have no structure
- This does not look disciplined – looks like a free for all
- But we need 6 more weeks to explore the infrastructure
- How can we do system verification testing (SVT) test for each iteration?

Rational Quality Manager project begins



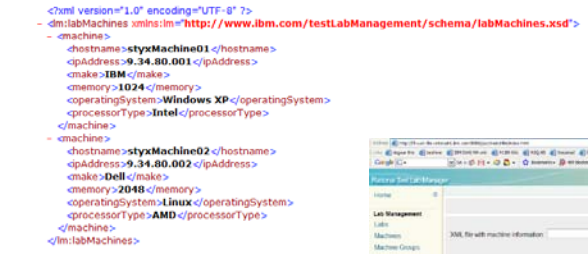
Test Plans and Test Cases in RQM

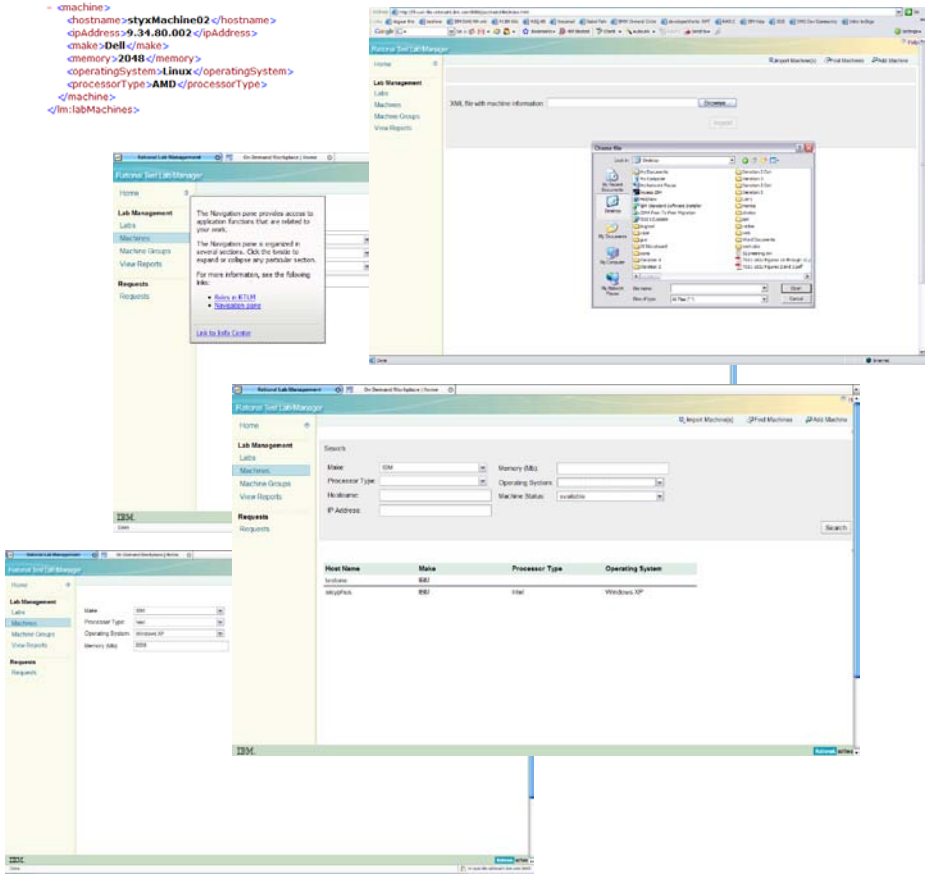


How the first two weeks went

From no code at all, to…

- ✓ Import xml file to populate database with machines
- ✓ Search for machines based off of any machine attribute
- ✓ Show a list of machines in a search results table
- ✓ Provide Context Sensitive Help
- ✗ Select a machine to see its details- missed del date by 6 hrs





What worked?

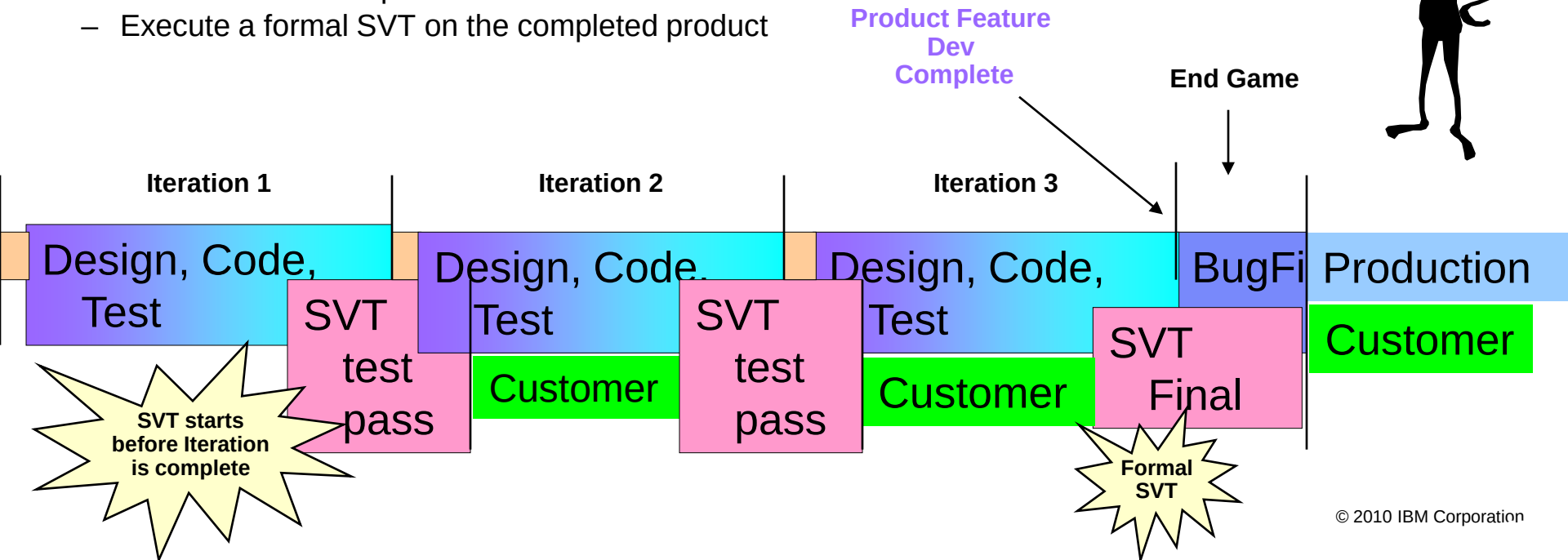
- Code containment within iterations
- Strong collaboration
- Cohesive problem solving: focus on the “now”
- Documentation for features was delivered
- Individual contributions were recognized and valued
- Install is a critical part of our product efforts
- SVT participation early in development cycle to provide feedback sooner

How System Verification Test Fits into Agile:

- Run on Milestone builds
- N Iterations = 1 Milestone
  - Aka I1 + I2 + I3 = M1, I4 + I5 + 16 = M2
- This allows for planning and test case development during a time when
  - delivered code is available to see running
  - some code is in progress and can be seen on a developer's machine
  - the features intended to be delivered can be seen on storyboards or in plan items.

Agile SVT approach

- Basic concepts
  - Execute system-level testing on all iterations with end to end stories
  - Testing may overlap development activities (either on current or next iteration) especially for performance and reliability runs
  - Focus on functional capability, usability, consumability, basic performance, and regression testing. While writing testcases using iterations, open defects. Be an early tester in a pair environment – even during prep activities.
  - Track successes, failures, defect status, and issues during the iteration test pass
  - Provide SVT feedback to development prior to all code deliveries to actual Customers
    * Defects
    * Assessment reports
  - Execute a formal SVT on the completed product



What does the SVT Schedule look like?

Key Project Milestones	Plan	Outlook / Actual	Completion / Comments
Plan DCP approval	03/10/09	03/10/09 A	Complete
SVT Test Plan Approval	03/20/09	04/09/09 A	Complete
Milestone Test Entry - M2/M3D1	03/23/09	03/23/09 A	Complete
Milestone Test Exit - M2/M3D1	04/17/09	04/17/09 A	Complete. 95% attempted, 88% completed.
Milestone Prep Test Entry – M3	04/20/09	04/20/09 A	Complete
Milestone Prep Test Exit – M3	04/24/09	04/24/09 A	Complete
Milestone Test Entry – M3	04/27/09		Green
Milestone Test Exit – M3	05/08/09		Green
SVT Prep Start	04/20/09	04/20/09 A	Complete
SVT Prep End	05/11/09		Green
Final DCUT	05/04/09		Green

The Team

- Issues
  - People find it hard to let go of original team members and to let new people in
  - Lack of bonding
  - Hard to see the BIG picture of how all of the team fits together
  - How do we know the customer will want this

- Solutions
  - No more “teams” – 1 team, areas have contributors
  - Test deliverables are reviewed by many contributors
  - Experience/Assessment reports are generated by test team for all to review their evaluation of the product’s health
  - People volunteer for comfort zones and people test out new areas and team mates mentor
  - Celebrations during a lessons learned meeting to discuss what went well and what to improve
  - Formation and participation in customer meetings and beta programs with real customers providing guidance

Accomplishments

- People are happy choosing what to work on and learning new things
- SVT has figured out the rhythm in the schedule for test development and execution
- Communication between Customer Team and the rest of the team is now on track - storyboards make a difference!!
- We are grateful for our awesome install team!

Take this home

- Short Iterations based off of estimates done by the team result in code delivered and tested that is consumable
- Complete and detailed storyboards completed before the iteration drives success
- Customer focus on requirements avoids rework and redirection later on

And this….

- The Team is comprised of Development, QE, Documentation, User Experience, Graphics, Install…
- QE loves being involved early and the product is delivered to SVT with higher quality when QE can influence the design of features
- SVT testing early eases test burden
  - Creates building blocks for suites and formal SVT entry…reuse test artifacts for regression testing (ROI)
- Celebrate your successes and acknowledge everyone's contributions