

Adapting Agile A TV Ratings Perspective

By Robert Black from Sharon Krossa Consulting
to Stanford Agile COP, 23rd January 2014

Adapting Agile to the Real World

- Compare Experiences of Methodologies
 - SCRUM-centric
 - Extreme Programming
 - Waterfall
 - RUP
 - PRINCE2
- What Problems are we Trying to Solve?
- What can we Learn?

Who Am I?

- Robert Black: rablack@stanford.edu
- Background
 - From the UK
 - Worked for Kantar Media for almost 10 years
 - Developing software used for TV Ratings in many countries
 - Managing a team of many Project Managers
 - I had the opportunity to try different methodologies with different project teams

Now

- Working for Stanford (and others) as a contractor
 - Through Sharon Krossa Consulting
 - Non-Stanford: rob.black@sharonkrossa.com
- Providing
 - Software Engineering Consultancy and Training
 - IT Management Consultancy and Training
 - Including for Drupal Web-site Projects
- Drupallers Drop-in Help
 - Cubberly Library 3-5 pm on 3rd Thursday of every month



I am Not going to Talk About...

- The Specifics of TV Ratings
 - Especially anything specific to Kantar Media

TV Ratings 101

- Seeks to Answer the Question “Who is Watching What and When?”
- Is a form of Market Research
 - A sample represents the population
 - “Who?” really means “How many of a particular demographic group?”
- Used for
 - Buying and Selling Advertising Airtime
 - Planning Programming
 - Designing Subscription Packages

Measuring TV Ratings

- Consistent Industry-wide Methodology
- Recruit a panel to represent the demographic distribution of the target population
- Collect TV Viewing Information from households
 - Paper Diaries
 - Electronic using “Meters” – hardware that connects to the TV and monitors usage
- Process the raw viewing data to generate the ratings
 - E.g. map broadcast frequencies to TV channels
 - Remove households with faulty equipment from the data

SCRUM

- Agile Lightweight (Software Development) Methodology based on Empiricism
 - Pillars: Transparency, Inspection, Adaptation
- Core Roles
 - Product Owner
 - Development Team
 - Scrum Master
- Ceremonies
- Sprints of 4-6 Weeks Without Changing Direction
- Ideal team size: 7

SCRUM – What Worked

- Very good for delivering custom products for a specific customer
- Frequent releases demonstrated progress
- Transparency accounted for money spent
- Able to Predict Future Progress
 - When the future was similar to the past
- Managed Quality over Many Iterations
- Especially Good for User Interface Development

SCRUM – What Didn't Work

- Projects with Requirements more Complex than can be managed by 1 Person
- Projects that Can't be delivered Incrementally
- Projects with a large non-software Component
- Projects of more than 20 People
- Projects with fewer than 4 People
- Projects that shouldn't be Continued
 - Success is realizing that the project is impossible
- Emergency Response

Waterfall

- The “original” Methodology
- Sequential
 - Analyze
 - Design
 - Construct
 - Test
 - Deliver
 - Maintain

Waterfall – What Worked

- Projects with low risk
 - Low complexity
 - Almost the same as the last project
- Projects with a high cost of iteration
 - Hardware
- Delivered what the customer needed instead of what the customer asked for

Waterfall – What didn't work

- R&D
- Complex software

Waterfall vs Scrum

- Scrum
 - Does better for software most of the time
- Waterfall
 - Handles the situation when iteration is impossible
 - In practice this means too expensive
 - Handles projects that are predominantly not about software
- Waterfall is almost never a “good” technique but sometimes Scrum is worse.

Extreme Programming (XP)

- A Set of Best Practices for Agile Software Development
- Practices Include
 - Pair Programming
 - Automated Unit Tests
 - Continuous Integration
- Self-organizing Teams
- Works with as few as 2 People
- Short Iterations: 1-2 Weeks
- Embedded Customer

XP – What Worked

- R&D
- A team with a natural leader who wanted to stay a programmer
- Extremely responsive to the customer asking for something
- Handled emergencies
 - Can change direction mid-iteration

XP – What didn't Work

- Problems with customers undermining best practice
 - Creating emergencies to “keep costs down”
- Sometimes what customers need isn't what they ask for
- Didn't scale well above about 6 people
- Requires programmers to be self-organizing. Did not work well with junior programmers.
- Self-organizing teams conflicted with best practice

XP vs Scrum

- Scrum
 - Does a better job of managing process
 - Can be used with less experienced staff
 - Does some management of the customer
- XP
 - Extreme best practice
 - Hands off approach suits some brilliant programmers
 - More nimble than Scrum
 - Handles emergency changes
- Scrum and XP are not mutually exclusive

What Problems are we Trying
to Solve?

Summary so far

- Scrum and XP work well for software because their feedback loops manage specification risk
 - Specification of software is hard
- Scrum manages process, developers and customers better than XP at the cost of longer iterations
- Waterfall's only real feedback loop is prototyping
 - However it manages more diverse risks than Scrum or XP
- Can we do better?

Rational Unified Process (RUP)

- RUP is an Artifact-driven Iterative Methodology
- Designed for Software Development
- Manages non-software components
 - Including managing their risks
- Methodology is very heavyweight unless you Tailor it

RUP – What Worked?

- Nothing!
- Methodology buried us in paperwork
 - Needs tailoring to remove unnecessary paperwork
- But knowing what is necessary requires experience
 - Expensive consultants!

PRINCE2

- Iterative Adaptive Methodology
- Developed and widely used by UK Government
- Several levels of feedback loops
- Includes separate “starting up” and “shutting down” phases
- Includes ongoing assessment of project viability
- Not directly suitable for Software Development
 - However, it is suitable for managing “suppliers” who use agile methodologies to deliver software

PRINCE2 – Strengths

- Manages all sorts of risk
- Integrates nicely with multiple Scrum teams

PRINCE2 - Weaknesses

- Heavyweight
- Needs all 6 key roles trained in PRINCE2
 - Not all key roles are necessarily different people
 - Steering Group (5 roles)
 - Executive (Decision maker. Represents the organization)
 - Senior User (Represents all users/customers)
 - Senior Supplier (Represents all suppliers, including Scrum Masters and Product Owners)
 - Project Manager
 - Project Assurance
 - Team Manager(s) (Scrum Product Owners)

What Did I Learn?

- Use Scrum with XP best practices for Software Development
- If there is a substantial non-software component then use a conventional Project Management Methodology layered over the top
 - e.g. PRINCE2 or PMBOK
- Make sure everybody is adequately trained
- If there are multiple Scrum teams, hire an architect and a business analyst