7 Habits of Highly Effective Agile Testing

Philip Lew
LinkedIn “Professional” Profile

• Phil Lew
  – Telecommunications consultant and network designer
  – Team Lead, Data warehousing product development
  – Software product manager, BI product
  – COO, large IT services company
  – CEO, XBOSoft, software qa and testing services

• Relevant specialties/Research
  – Software quality process improvement
  – Software usability evaluation
  – Software quality in use / UX design
Cone of Learning (Edgar Dale)

After 2 weeks we tend to remember...

- 10% of what we READ
- 20% of what we HEAR
- 30% of what we SEE
- 50% of what we HEAR and SEE
- 70% of what we SAY
- 90% of what we both SAY and DO

Nature of Involvement

- Verbal Receiving
- Visual Receiving
- Receiving / Participating
- Doing

Session Spirit and Expectations

• I won’t read the slides…
• Slides for you as a take-away (many)
• Maybe different than your handouts
  – Always thinking of new examples and ideas, you can write me and I’ll send these to you
• Interactive
  – Lots of chances to ask questions and learn
• You’ll learn more today than software stuff.
Foundation of 7 Habits

• Stephen Covey’s book in 1989 almost 30 years ago, “Seven Habits of Highly Effective People”

• Just starting my career, I applied many of the principles to not just my work life, but life as a whole.

• Recently read his book again and sharpened “my saw” in the process. With these principles in mind, I developed this talk.
The Original 7 Habits
by Stephen Covey

1. Be Proactive
2. Begin with the End in Mind
3. Put First Things First
4. Think Win-Win
5. Seek First to Understand, Then to Be Understood
6. Synergize
7. Sharpen the Saw
The 4,200-foot long suspension span of the Golden Gate Bridge was the longest span in the world at the time of its construction in 1937. On time, still standing.
Software is New and Constantly Changing

1. Many failures
2. Changes in technology (mobile, cloud) and SaaS business models
3. Demand changes in the way we work

- Smaller teams
- Faster iterations
- Listening to the user
  - Continuous beta
  - Data collection
- Communication
- Analysis, adaption and improvement
Being Successful or Not with Agile
After Working With Many of Our Clients

• I’ve found a few traits that are common across all or at least most
  – Either in their success
  – Or in their failure

• Most of these are not rocket science

• Some have nothing to do with technology
Agile Failures – Why?

Let’s Vote, Pick 3

- Process Inconsistency
- Lack Test Automation
- Retrospectives Not Valuable
- Agile Fall (ScrummerFall)
- Agile Doing – Not Being
- Requirements Churn
- Distrust
- Resistance to Change
- Lack Customer-User Understanding

Agile Problems
“Be Proactive” at Being Successful and Prevent Failure!

Top Reasons For Agile Failure

- Process Inconsistency: 19%
- Agile Doing-Not Being: 18%
- Agile Fall: 13%
- Resistance to Change: 13%
- Requirements Churn: 10%
- Retrospectives Not Valuable: 8%
- Lack of Customer/User Understanding: 8%
- Lack of Test Automation: 6%

@philiplew @xbosoft © 2017 XBOSoft, Inc.- All Rights Reserved.
Habit 1
Focus on Efficiency and Effectiveness

“Smart and Get Things Done” (from Joel on Software)

- High level of test automation with continuous integration and delivery
- Fast feedback systems
- Automate environment and data management
- Do just “enough documentation”
- Prioritize your efforts – “Put First Things First”
- Manage technical debt and risk
Plan for Technical Debt

• Plan for it – It will happen, how to “Be Proactive” and deal with it?
• Agile teams should have a “fix bugs first” or bug thresholds to prevent a large backlog of bugs
  – Reduce ‘time to fix’
• Periodic stabilization or ‘fix defects’ sprints should be planned
• With proper metrics, testing can help measure and manage technical debt
  – What metrics could we use for technical debt?
Have Criteria for Documentation or NOT

“Agile methods derive much of their agility by relying on the tacit knowledge embodied in the team rather than writing the knowledge down in plans.”

Barry Boehm

What would be your criteria for having documentation?
Habit 2

Treat the User as Royalty

- Getting user stories right—the main input for testing
  - Understand context and domain
  - Ask questions or plan on lots of rework
- Prioritize product quality, user experience and value to the client and user
  - Reduce list of things to do—What criteria do you have?
- Conduct usability testing as early as possible
  - Use mock-ups as throw away tests
  - Hallway usability testing
Understand the User (Empathy)

Test early and often with users and customers

• Task analysis
  – 80/20 rule

• Real usage and observation
  – Don’t wait till the end

• Each iteration should provide valuable information going forward
  – Don’t race to the finish line, you may be going in the wrong direction!
Habit 3
Maintain an Improvement Frame of Mind

• Continuous improvement through tracking and measurement
• Trusting and meaningful retrospectives
• Develop agile practices with constant review and improvement as part of the process
• Maintain a sustainable pace that leads to sustainable improvement
Metrics Should

• Affirm and reinforce your Agile goals and principles
• Show trends not numbers
• Be a basis for conversation
  – WHY
• Be easy to collect
• Help determine excellence and improvement (or not)
• Be easily and openly viewed every day by all
Metrics

• Team understanding and acceptance of what is measured:
  – Support building better software
  – Not the end, but a beginning of a discussion to get better

• Enabling:
  – Evaluation of progress
  – Redirection of priorities
  – Insights on if changes worked
Agile Metric Areas

• Burndown progress
• Velocity trend
• Work item status
• Team work loads
• Testing results
• Delivery

What metrics do you use?
Meaningful Retrospectives
Use It to Sharpen the Saw

• Held at the end of each iteration
• Team reflects on what happened in the iteration and identifies actions for improvement
• Team-driven, and team members should decide together how the meetings will be run and how decisions will be made about improvements.
• For continuous improvement, the Agile retrospective is one of the most important of Agile development practices.
(A) Retrospective Framework

Don’t just “talk” - Begin with the End in Mind

- Initiate
- Collect info
- Generate Insights
- Decide on Action
- Close
Retrospective Framework Example < 1 Hour

1. **Initiation** - Get the team ready to engage with a participative warm-up activity that relaxes and engages (5 minutes)

2. **Collect information** - Create an overall team view of what happened during the iteration (10 minutes)

3. **Generate insights and action items** (15 minutes)
   - What worked well for us? Or what should we continue doing?
   - What did not work well for us? What should we stop doing?
   - What actions can we take to improve our process going forward?
     What should we start doing?

4. **Decide on Action** - identify highest priority items for the next iteration and measures to determine completion (15 minutes)

5. **Close** - Reflect on the effectiveness of the retrospective and how to improve it (5 minutes)
   - Appreciate accomplishments of the team and individual interactions (5 minutes)

Does anyone have another example?
Retrospective’s Value Depends on Trust

• An atmosphere of honesty and trust so team members can share thoughts.

• Norman Kerth's work, his prime directive states:
  
  • "Regardless of what we discover, we understand and truly believe that everyone did the best job they could, given what they knew at the time, their skills and abilities, the resources available, and the situation at hand."

• Permission to be gloom and doom
Trust

We feel trust. Emotions associated with trust include companionship, friendship, caring, love, comfort.
Trust is Created by Action

• Here are a few ways to create trust in our daily lives *through our actions toward one another and our clients*.

1. Predictability
2. Exchange
3. Reciprocity
4. Vulnerability
1: Predictability

• **Definition 1**: Trust means being able to predict what other people will do and what situations will occur.

• If we can surround ourselves with people we trust, then we can create a safe present and an even better future.
2: Exchange

• **Definition 2**: Trust means making an exchange with someone when you do not have full knowledge about them, their intent and what they are offering to you.

• Interactions with exchange are basis for all relationships.

• Trust in value exchange if we do not know whether what we are receiving is what we expect.
3: Reciprocity

• **Definition 3**: Trust means giving something now with an expectation that it will be repaid, possibly in some unspecified way at some unspecified time in the future.

• Trust is important here because otherwise we are giving something for nothing. The delay adds uncertainty that requires trust.

• “I give you something now with the hope of getting back some unspecified thing in the unknown future.”
4: Vulnerability

• **Definition 4**: Trust means enabling other people to take advantage of your vulnerabilities (weakness)—but expecting that they will not do this.

• When we trust other people, we may not only be giving them something in hope of getting something else back in the future, we may also be exposing ourselves in a way that they can take advantage of (us) our vulnerabilities.
What Do All These Elements of Trust Have in Common?

- Predictability
- Exchange
- Reciprocity
- Vulnerability

Trust is developed by reducing or removing uncertainty.
Habit 4

Wants to be Agile

Develop Agile

• You can’t do Agile unagile you are
High School Be-Do-Have

Be

Do

Have

© 2017 XBOSoft, Inc.- All Rights Reserved.
Agile (Doing) Best Practices

What Agile Best Practices are Actually Applied?

• 86% have short iterations
• 84% use daily scrum meetings
• 80% have a product owner
• 78% gain early and frequent feedback
• 68% use burn-up/burn-down charts

Agile Be-Do-Have

Being - People
• Communicative
• Collaborative/Cooperative
• Flexible and willing
• Knowledgeable-multi
• Initiative/responsible
• Responsive

Doing - Process
• Iterative (sprints)
• Daily standups
• Face to face communication
• Post mortem – end of sprint
• Delivery meeting – end of sprint
• Planning meeting – before sprint
• Self organizing

Having - Results
• Speed
• Quality

Focus Measurements Here

© 2017 XBOSoft, Inc.- All Rights Reserved.
How to Support and Enable agile Being

• Maintain small teams
• Promote cross functional stacked teams
• Emphasize collaboration versus 1 way communication
• Let the quiet people talk
• Develop a shared understanding
• “Think Win – Win” and “Synergize” through your team culture
Keep Teams Small – Size Matters

“Synergize”

• Leads to cross functional teams that are full-stack and autonomous won’t get blocked
• Promote efficient communication and collaboration so that team members do not block each other and duplicate their work;
• Can integrate feedback after each iteration and make changes easier and faster than larger teams
On the Other Hand, Big Teams:

- Cause groups to form around a discipline – can inhibit cross-discipline collaboration.
- Can lead to disrupted cadence, diluted shared understanding, and dispersed focus.
Communication is in our Evolutionary DNA

With other things equal, Teams that communicate better will win.
### Communication versus Collaboration

<table>
<thead>
<tr>
<th>Two-Way</th>
<th>Gossip and small talk</th>
<th>Collaboration</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-Way</td>
<td>For your information</td>
<td>Broadcast</td>
</tr>
</tbody>
</table>

- **Collaboration**: Two-way communication aimed at achieving specific business objectives.
- **Those that communicate AND collaborate the best will WIN**

http://bizblog.blackberry.com/2013/11/mind-link-collaboration/

[Image 0x0 to 720x540]
Communicate AND Collaborate

- Email is necessary communication, but it is not equal to collaboration.
- Utilize high bandwidth communication; face to face, Skype, telephone, whiteboards, video, etc...
- Create process that includes communicating as part of the process.
  - Don’t write documents to avoid talking, CYA docs-emails
  - Decide what docs will be useful in the beginning
Iteration Goes Hand in Hand with Collaboration

• Incomplete and imperfect designs should be welcomed as imperfection that leads to collaboration.

• Recognize that iteration leads to better design.

• “Working” software means it works enough to get feedback, not perfection.

• Use tools such as prototyping tools as a way to collaborate.
Promote a Shared Understanding

When you hire new team members

• Don’t assume new guys have the same shared ‘understanding’ that you’ve developed over time.

• Pause to initiate new team members, answer questions, and appreciate new input that could and should be different.

• Provides opportunity to reset/reinforce the entire team’s understanding.
Promote a Shared Understanding

“Seek First to Understand Then to Be Understood”

• Shared understanding is the key to Agile.
  – Don’t make assumptions on what is understood or not.
  – Ask: what do you mean?

• Apply XP programming techniques to bring new hires up-to-speed quickly, e.g., pair programming.

© 2017 XBOSoft, Inc.- All Rights Reserved.
Facilitate a Team Culture

• Maintain team composition consistency.
  – People coming in and out of the team can disrupt cadence, flow, and collaboration.

• When someone is missing (for any number of reasons) for a day or two, that’s not collaboration.
  – Working in isolation is necessary sometimes, but should be very short duration. Make work and progress (or no progress) instantly visible to all.

• Changing roles is good to break up a routine
  – learn new things and provide future contingency/risk mgmt, but can also bring productivity and cadence down.
Habit 5
Think Tasks Not Roles

• No ‘testing’ phase, but many testing activities and work items
• Flexibility in roles but with clear responsibilities and task assignments
  — You can wear a different hat?
  — What hats can you wear?
• QA is part of the team from start to finish
Testing Tasks are a Critical Part of the Plan

• Testing is not an afterthought
  – Clarify stories with customer and end user
    • Think from their point of view
  – Identify testing activities to support development
    • White box testing
    • Integration testing
  – Backlog grooming
    • Improve and clarify stories

• Testing is part of each story work item
Tom Brady Can Play Linebacker

- Go where you’re wanted and needed
- Use the skills you have where they are most needed to help the team win
- AND... GET MORE SKILLS – “Sharpen the Saw”

@philiplew @xbosoft
Tester Pairing

• With test domain experts
  – Security
  – Performance
  – Usability
  – Automation

• With other testers
  – “Extreme Testing” -- XT (XP)

• With programmers
  – White box
  – Programming logic versus testing logic
Make Responsibilities Clear

• Lots of chefs are great but...
• Divide the work according to type of chef (dessert, meat, pastry)
• Need clear decision makers in each discipline and have specific roles (can also be rotated)
• Weakest link can cause the team to fail
• Talent must match the needs of the team and the team must have the ability to replace talent.
• Identifying the weak links should be obvious via working in close proximity and transparency.
Habit 6
Focus on the Customer

• Collaborate with the customer
  – Help them understand what their requirements are!
  – Help them prioritize
  – Extract, elicit examples → user stories, test cases
  – ‘Mind the gap’ – be a bridge between the developers (tech language) and the users/business analysts (business language)
  – “Help Me Help You!”
“Help Me Help You”  
Jerry Macguire  
Video

1. Sports agent and American football player
2. Football player is not playing well
3. The sports agent is trying to negotiate a new contract for him
4. But his poor and unmotivated play is not helping
Work Closely with Customers

• “Help You Help Me” and “Help Me Help You”

• Ensure you have a skilled and dedicated product owner from the business
  – Why bother with Agile if the business stakeholders are not ACTIVELY involved?
  – They should be leading the way, not dragged along. They should be prioritizing requirements.
  – Focus on minimum viable product to get early team success.
Agile is Designed for the Customer

• Recognize that the most difficult part about requirements that Agile was designed to solve is that of missing information.
• The purpose of Agile is to gradually fill in missing information while development progresses within each sprint.
• **Prioritize** on what serves the product, customer and user best.
Habit 7
Think Long Term

• “Begin with the End in Mind”
  – Improve the software
  – Deliver the ‘best’ software you can.
  – Deliver working software

• Keep the user in mind—Satisfy the user

• Do what is sustainable, or like any race where you run to fast, you’ll quit.
Work Delivered at a Constant Pace

• Recognize it's a long term effort, a marathon, not a sprint
  – The amount of work delivered per sprint is constant
    • Add and subtract
    • Pace and expectations
    • Don’t compare yourself with others

• Improve incrementally, not just the software but you, and your team and how you work together – Continually “Sharpen YOUR Saw”
Develop a Long Term Agile Diet

• After you start changing your diet, don’t let the old habits creep back in.
• Maybe takes longer than “21 days”.
• It’s just like a diet. Losing weight for a short period is not the goal. You want to keep it off for life. It's the beginning of a new lifestyle.
• “Agile makes sense” but so does eating fewer desserts. Hard to change long term.
Key Components of the Agile Diet

• Short iterations; don’t have to be 2 weeks
  – More iterations, the better – if the overhead is bearable.
  – How often do you weigh yourself?
• Integrate feedback after each iteration retro and course correct.
• Daily scrum meetings with real communication
  – everything is not ‘normal’ or ‘ok’
• Committed and involved product owner
  – Early and frequent feedback from real customer as well
Key Components of the Agile Diet

• Don’t forget downstream activities (remove barriers between development, test, and operations)
  – Continuous integration
  – Continuous delivery

• Continuous improvement
  – Measurement and metrics

• Have productive retrospectives
  – We ate too many desserts, let’s work on this!
Agile New Year’s Resolutions

Have I Told You ANYTHING you didn’t know?

When the doctor tells you to lose weight, is it something you don’t know? (Probably not.)

I hope I’ve reinforced what you know or shed a new angle on things you’ve heard.

(Yes, eating fried foods is not good for you!)
Successfully ‘Being agile’

• Agile is not a technique
• Agile is not a technology
• You can be agile without a book, without a coach, just like you can lose weight if you want to, without Weight Watchers

• With a diet, you want a lifestyle change
  – Not just a 3 week effort, but for long term health
  – Sometimes you have set backs

• With Agile, you want a work-style change
The 7 Habits of Highly Effective Agile Testing

1. Proactively be efficient and effective
2. Treat the user as royalty
3. Maintain an improvement frame of mind
4. Be agile, then do Agile
5. Think tasks not roles
6. Focus on customer
7. Think long term
Thanks!

For more, download:

Agile Metrics Whitepaper
Agile Test Plan

http://xbosoft.com/resources/

Philip Lew
@philiplew
philip.lew@xbosoft.com