What’s that Smell?
Early Warnings of Agile Project Issues

By Kevin Aguanno, B.A., IPMA-B, PMP, MAPM
Executive Project Manager, Agile Coach, Author
Your Presenter: Kevin Aguanno

- 20+ years of PM experience
- 20+ published books, audiobooks, DVDs, and CD-ROMs – most on agile and PM-related topics
- IBM Certified Executive PM
- IPMA Certified Senior PM (IPMA B)
- IBM AIS Agile Centre of Competency Lead
- IPMA-Accredited PM Competency Assessor for Canada and USA

kevin@AgilePM.com
My Approach to Agile Project Health Checks
Agile Project Health Check
Root Cause Analysis: Late Code Delivery

- Iterations 1 & 2 Late
- Iteration 3/4/5 Forecast Late
- Work Underestimated
- Misunderstood Requirements
- High Level of Code Defects
  - Communications Management Issues
  - Coding Before Requirements & Design
- Vendor Lacks Domain Expertise
- Not Following SW Dev‘mnt Best Practices

Root Cause
Root Cause
GenXus Agile Project Health Score™
As at Aug. 2011

Project XXXXX Agile Health
as at Aug. 2011

Overall Project Score: 5.1 / 10

The improvement since Dec. 2010 is attributed to the corrective actions taken since the start of 2011.
Problems on Agile Projects can be Grouped into 3 General Areas

1. Problems with Team Engagement
2. Process Adherence Problems
3. Performance Problems
Problems with Team Engagement
Missing Team Members in Daily Meetings

**Symptoms**
- Team members not present for daily meetings or present but not participating
- Perfunctory reports
- Disruptive behaviour
- Team members complain that daily meetings are not adding value

**Impact**
- Decreased morale
- Decreased teaming
- Disrupted rhythm
- Decreased visibility into issues

**Remedies**
- Establish a clear rhythm
- Model proper behaviour
- Explain the benefits, persuade, and negotiate
- Change meeting time and location
- Embrace technology to overcome time/distance issues
- Reorganize the team/replace resources
- “Dis-invite” external stakeholders and keep the meetings only for core team members

Source: Mark W. Randolph
Team Members Feel that Daily Meetings are for the Project Manager

**Symptoms**
- Team members complaining they are not seeing the point (value) of the meetings
- They are in a hurry to “get it over with”
- Poor meeting attendance
- Team members take over the meeting to deal with technical discussions

**Remedies**
- Explain what benefits the team members can get out of these meetings
- Motivate and persuade
- Don’t let the PM be the focus of the meeting
- Adopt a “servant-leader” management style

**Impact**
- Poor coordination (leads to waste)
- Poor visibility into issues
- Increased # of surprises

Source: Mike Cohn
Overly-Specialized Job Roles on Team

**Symptoms**
- Work is “handed off” from one role to another with clear boundaries in responsibilities
- Little cross-role collaboration
- Class-based (hierarchical) role distinctions
- Team members have lots of idle time waiting for upstream activities to complete so they can start

**Impact**
- Inefficient allocation of resources
- Sub-optimal productivity / velocity
- Little knowledge transfer or collaboration

**Remedies**
- Set up performance evaluation system that values team success over individual success (i.e. each is measured individually on the success of the weakest member)
- Clarify overlaps in skill sets between roles
- Seek to build “generalizing specialists” in the organization
- Revised formal job descriptions for agile team members
- Get HR or union reps onside with the plan

Source: Mike Cohn
Testers will Not Integrate with the Team

**Symptoms**
- All testing treated as an independent activity
- Independence of testers required as part of governance model
- Testers not allowed to interact closely with developers as it may interfere with their independence
- Testers report into a separate org. than the rest of the team

**Impact**
- Lower overall productivity
- Lower quality due to later feedback to developers
- Less efficient testing activities

**Remedies**
- Make testing a strategic competence in the development activities, not an afterthought
- Explain how testing metrics can be used to improve development processes, especially with continuous testing
- Mix “constructive” and “destructive” personalities on the development team
- Borrow a tester from the independent team to help with testing during development and then send them back to the independent team for their verification testing.

Poor Attendance at Retrospectives

**Symptoms**
- Project sponsor or team members not attending the end-of-iteration retrospective
- If attending, participants arrive late or leave early
- Those attending physically are not participating in the meeting

**Impact**
- Missing opportunities to improve the project processes
- Other team members soon start to lose respect for the process as well
- Business loses trust that the team is fixing problems

**Remedies**
- Schedule the meetings earlier in the day (Friday afternoons = bad idea)
- Schedule meetings over lunch and bring food
- Take the process seriously by acting aggressively on opportunities for improvement raised during the sessions
- Roundtable approach asking each person for some input (be careful to get low-participating individuals to speak first/early, else they just say “I agree with the others”

Source: Mark Levison
Poor Teaming

Symptoms
- Team members not collaborating or sharing knowledge and experience
- Team members not communicating (i.e. are working in isolation)
- Team members engaging in “hard” (or “cold”) handoff approaches
- Team members not helping each other complete their activities

Impact
- Lower productivity / velocity
- Lower morale (possible infighting)

Remedies
- Measure team members’ performance based on the group achieving its objectives
- Reward collaboration, knowledge transfer/mentoring, and sharing of existing assets
- Focus on teaming issues during retrospectives and planning sessions
- Create more flexible job descriptions

Source: Henrik Kniberg
Dominating Team Member

Symptoms
- One “voice” dominates discussions
- One team member telling others what to do
- One person providing estimates for the team
- Group refuses to make a decision without input from the dominant member
- Bullying

Impact
- Missing valuable input/insight from introverted team members
- Lower morale
- Lower productivity

Remedies
- Use techniques like Planning Poker™ to get the true thoughts and opinions of every team member
- Stronger facilitation from the PM or ScrumMaster to moderate the impact of the dominant team member – draw other team members into the discussions
- Enforce a disciplined, collaborative estimating approach
- In worst-case scenarios, consider replacing the dominant resource

Source: Mark Wainwright
Team is Reluctant to Estimate Backlog Items

**Symptoms**
- Team not using a disciplined estimating and planning approach to provide forecasts to the business
- Backlog maintenance only performed by project sponsor without team involvement
- Team keeps claiming they don’t have enough information to estimate the backlog

**Remedies**
- Training on estimating practices
- Stronger facilitation of estimating meetings
- PM to shield the team from the negative responses from the business due to incorrect estimates
- Make it OK to be a little wrong sometimes – always right means you have too much contingency in your estimates
- Use a two-tiered backlog (primary and secondary) to separate items ready for estimating from those with significant unanswered questions

**Impact**
- Team can’t provide a reasonable view of final schedule or budget
- Poorly-set expectations lead to surprises for the business

Source: Rown Bunning
Problems with Agile Process Adherence
# Loss of Rhythm

## Symptoms
- Iteration planning meetings are skipped or are poorly attended
- Iteration lengths are inconsistent or are changed mid-iteration
- Daily team meetings are poorly attended or meeting times vary
- Daily team ritual is drifting

## Remedies
- Ensure the agile PM has been adequately trained
- Ensure expectations for meeting participation are understood
- Ensure that daily meetings are short and do not drift into problem solving – enforce good practice
- Protect the team from outside interference
- Set consistent iteration lengths

## Impact
- Decreased productivity
- Decreased forecast accuracy
- Decreased morale

Source: Mark W. Randolph
Lack of Technical Standards

**Symptoms**
- Increasing frequency and size of refactoring work
- Large number of high-severity defects when trying to integrate code

**Remedies**
- Implement daily builds
- Implement nightly automated test runs
- Have the team agree on a number of standards, then have them refresh their memory on those standards at the start of each iteration
- Perform code inspections to enforce standards (pair programming)

**Impact**
- Without standards, collective code ownership becomes problematic
- Greater retesting and defect repair costs

Source: Mark Levison
Lack of Process Improvement

**Symptoms**
- The same mistakes are being repeated over and over
- Productivity (velocity) not improving in early iterations

**Impact**
- Missed opportunities to improve productivity, costs, schedule, etc.
- Needless waste

**Remedies**
- Formalize the use of retrospectives at the end of each iteration
- Include the business sponsor in the discussions (or at least have a separate lessons learned discussion with the sponsor)
- Have the group decide on what action to take (if any) for each lesson learned
- Output of the retrospective may be new project backlog items, or a revised approach during upcoming iteration planning sessions

Source: Henrik Kniberg
Excessive Executive Pressure

Symptoms
- Sponsors or other stakeholders micromanaging team members
- Interference from the business during iteration execution activities
- A group of sponsors co-owning the project backlog with no single person in control
- Team members working on items not in scope for the current iteration due to stakeholder request

Impact
- Lowered overall productivity
- Lost time due to interruptions
- Inefficient work scheduling
- Lower morale

Remedies
- Enforce the rule that business stakeholders cannot redirect the team during an iteration unless there is a “stop work” situation
- Don’t put the team between stakeholders who are in disagreement about priorities; rather, have a strong PM/facilitator encourage them to work out their differences first and then bring the compromise position to the team
- Try to encourage the business to assign a single overall “owner” of the project

Source: Henrik Kniberg
Interfering External Stakeholders

**Symptoms**
- Non-team members speak up in daily team meetings
- Priorities switched outside of iteration planning meetings
- Team not allowed to make purely technical decisions without outside approval
- Project backlog is not maintained or is ignored

**Impact**
- Decreased productivity
- Decreased forecast accuracy
- Decreased morale

**Remedies**
- Enforce good meeting practice
- Train stakeholders at project start – highlight rules
- Reinforce impact of interference in retrospectives
- Remove team (physically) from proximity of stakeholders – “out of sight, out of mind”
- Keep external stakeholders out of the daily team meetings

Source: Mark W. Randolph
Disagreement on Definition of “Done”

**Symptoms**
- Over-engineered or over-built solutions
- Conflicting understanding between team members or between the team and stakeholders of acceptance criteria
- Stakeholders arguing over final acceptance criteria
- Undefined non-functional requirements

**Remedies**
- Have business analysts capture the business acceptance criteria at the time they capture the requirements
- Provide the acceptance criteria to the developers before they design and build their features to be used as design input
- Test-Driven Development
- Discuss up front with the business the strategy around documentation, governance presentations, etc.

**Impact**
- Wasted business investment by “gold plating” the solution
- Rework as acceptance criteria are changed throughout the project

Source: Henrik Kniberg
Tasks Assigned by PM Rather than Self-Selected by Team Members

**Symptoms**
- PM / Scrum Master is assigning tasks to team members
- Team members dissatisfied with the task assignments they have during an iteration

**Remedies**
- Have the team members prepare their own task list (WBS) for each iteration
- Have the team members agree amongst themselves who is going to do each task
- The role of the PM in these activities is to facilitate the process / discussions and to document the results, NOT to make any decisions

**Impact**
- Poor morale
- Possibly sub-optimal productivity
- No team support for plan and commitment

Source: Mike Cohn
Performance Problems
Lack of Progress

Symptoms
- Velocity is weak and has not improved over the first few iterations of the project
- Project burndown chart shows a flat line or at least one with a very weak slope
- Average velocity over past few iterations is far below the original planned velocity

Remedies
- Perform a root cause analysis to see what is slowing progress
- Add resources or bring in expert mentors to overcome specific challenges
- “Fast track” the iterations, performing the requirements and design work one iteration in advance of the build/test activities, allowing more time for the BAs and Architects to perform their work

Impact
- Increased schedule, increased budget, or reduced scope
- Lower team morale
- Sponsor distress

Source: Mark W. Randolph
Persistent Wild Fluctuations in Iteration Burndown Charts

Symptoms
- Iteration (Sprint) burndown charts show a trendline jumping upwards and downwards erratically rather than a relatively smooth downward slope

Remedies
- Perform a root cause analysis to identify the causes of the jumps
- If caused by surprising complexity, then “Fast Track” the work, having the requirements and design work occur one iteration ahead of the development work – this allows more time for requirements and design activities
- Provide experienced, senior mentors to team members during their iteration planning sessions
- Provide more training on sound estimating practices

Impact
- It is unclear whether or not the team will complete the iteration’s work on time
- Forecasting is more risky

Source: Mike Cohn
Consistently Missing Iteration Commitment

**Symptoms**
- Velocity measured at the end of each iteration is consistently less than 100% of plan
- High number of outstanding defects at the end of each iteration
- New defects at the end of an iteration are preventing the demonstration of newly-developed features

**Impact**
- High risk of going over budget, over schedule, or not completing enough scope to satisfy the business case
- Loss of trust with the business

**Remedies**
- Ensure a disciplined approach is taken to recalibrating the release plan based on actual observed velocity from the previous iteration
- Review development practices to look for ways to address quality issues (part of ongoing continuous improvement activities)
- Have the team make less aggressive commitments
- Ensure that there is no significant work taking place that is not showing up on the project backlog (i.e. “hidden” work that is consuming team productive hours)
- Minimize interruptions and multi-tasking

Source: Henrik Kniberg
Growing Technical Debt

Symptoms
- A statistical control chart analysis of velocity actuals shows a “heartbeat” pattern
- Lots of new stories being raised for refactoring
- Actual velocity trending downwards over time as things are taking longer and longer than expected to complete
- Defects not being addressed in a timely manner

Impact
- Surprise budget/schedule/scope/quality issues late in the project.

Remedies
- Discuss impacts (trade offs) of early or late scheduling of refactoring work with project sponsor
- Track defect open vs. close ratios, defect injection rates, and similar metrics to determine size of problem
- Consider a defect burndown chart to see if defects will be complete by end of project
- Insert stories into the backlog and release plan to assess the adherence to non-functional requirements, revealing the extent of the issue
- Add (or reprioritize) iterations so that there are “empty” iterations at the end of the project to absorb slippage.

Source: Henrik Kniberg
SUMMARY

- When assessing agile project health, look at all three areas:
  - Problems with Team Engagement
  - Process Adherence Problems
  - Performance Problems

- Watch for the symptoms of problems – they can be your early warning signs

- When spotting symptoms, determine the underlying problem and take corrective action early, before little problems become big ones
Questions?

- Kevin Aguanno (your speaker) is available for consultation at kevin@AgilePM.com.
- He is the author of over 20 books, audiobooks, DVDs, and CD-ROMs related to this subject matter:

  **Books:**
  - An Introduction to Agile
  - Health Methodology Adoption
  - Managing Agile Methods
  - Agile Test Management
  - 101 Ways to Reward Those Members Per $0 (For Less)

  **CD-ROMs:**
  - Agile Planning GAME

  **Audiobooks:**
  - Agile Project Management Using Scrum
  - Managing in the Face of Ever-Changing Requirements
  - Extreme Scrum
  - Agile Adoption: Making it Work
  - Mastering the Art of User Stories for Quality, Performance, and Testability

  **DVDs:**
  - Disrupting Commonly Held Byths About Agile
  - Introduction to Agile Project Management