Agile Practice Guide

Errata – 3rd Printing

N**OTE:** The following errata only pertain to the **first and second printing** of the *Agile Practice Guide*. In order to verify the print run of your book (or PDF), refer to the bottom of the copyright page (which precedes the Notice page and Table of Contents). The *last* numeral in the string beginning "10 9 8" etc. denotes the printing of that particular copy.

Minor editorial changes have been made to the text and figures. Notable corrections are listed below.

Agile Practice Guide

Page	Correction
16	Last paragraph, last sentence. Changed to: In order for the project to become reliably possible, it needs one of the *uncertainty* variables to be contained.
46	Section 4.3.6, 2nd paragraph, last sentence. Changed to: Therefore, companies are designing their offices to balance common and social areas with quiet areas or private spaces where individuals can work without being interrupted *(sometimes called "caves and common")*.
58	Table 5-1, row 8 (Team struggles with obstacles), col 2, last sentence. Changed to: Sometimes, the team needs to escalate *obstacles* the team or servant leader has not been able to remove.

These iterative, incremental, and agile approaches work well for projects that involve new or novel tools, techniques, materials, or application domains. (Refer to Section 3 on Life Cycle Selection). They also work well for projects that:

- Require research and development;
- Have high rates of change;
- Have unclear or unknown requirements, uncertainty, or risk; or
- Have a final goal that is hard to describe.

By building a small increment and then testing and reviewing it, the team can explore uncertainty at a low cost in a short time, reduce risk, and maximize business value delivery. This uncertainty may be centered on suitability and requirements (is the right product being built?); technical feasibility and performance (can this product be built this way?); or process and people (is this an effective way for the team to work?). All three of these characteristics—product specification, production capability, and process suitability—typically have elements of high uncertainty.

However, iterative and incremental approaches have their limits of applicability. When both technology uncertainty and requirements uncertainty are very high (the top right of Figure 2-5), the project moves beyond complex to chaotic. In order for the project to become reliably possible, it needs one of the uncertainty variables to be contained.

4.3.6 TEAM WORKSPACES

Teams need a space in which they can work together, to understand their state as a team, and to collaborate. Some agile teams all work in one room together. Some teams have a team workspace for their standups and charts, and work on their own in cubicles or offices.

While companies are moving toward open, collaborative work environments, organizations also need to create quiet spaces for workers who need uninterrupted time to think and work. Therefore, companies are designing their offices to balance common and social areas with quiet areas or private spaces where individuals can work without being interrupted (sometimes called "caves and common").

When teams have geographically distributed members, the team decides how much of their workplace is virtual and how much is physical. Technology such as document sharing, video conferencing, and other virtual collaboration tools help people collaborate remotely.

Geographically distributed teams need virtual workspaces. In addition, consider getting the team together in person at regular intervals so the team can build trust and learn how to work together.

Some techniques to consider for managing communication in dispersed teams are *fishbowl windows* and *remote pairing*:

- Create a fishbowl window by setting up long-lived video conferencing links between the various locations in which the team is dispersed. People start the link at the beginning of a workday, and close it at the end. In this way, people can see and engage spontaneously with each other, reducing the collaboration lag otherwise inherent in the geographical separation.
- Set up remote pairing by using virtual conferencing tools to share screens, including voice and video links. As long as the time zone differences are accounted for, this may prove almost as effective as face-to-face pairing.

Form teams by bringing people with different skills from different functions together. Educate managers and leaders about the agile mindset and engage them early in the agile transformation.

Table 5-1. Agile Pain Points and Troubleshooting Possibilities

Pain Point	Troubleshooting Possibilities
Unclear purpose or mission for the team	Agile chartering for purpose—vision, mission, and mission tests
Unclear working agreements for the team	Agile chartering for alignment—values, principles, and working agreements
Unclear team context	Agile chartering for context—boundaries, committed assets, and prospective analysis
Unclear requirements	Help sponsors and stakeholders craft a product vision. Consider building a product roadmap using specification by example, user story mapping, and impact mapping. Bring the team and product owner together to clarify the expectations and value of a requirement. Progressively decompose roadmap into backlog of smaller, concrete requirements.
Poor user experience	User experience design practices included in the development team involve users early and often.
Inaccurate estimation	Reduce story size by splitting stories. Use relative estimation with the entire team to estimate. Consider agile modeling or spiking to understand what the story is.
Unclear work assignments or work progress	Help the team learn that they self-manage their work. Consider kanban boards to see the flow of work. Consider a daily standup to walk the board and see what work is where.
Team struggles with obstacles	A servant leader can help clear these obstacles. If the team doesn't know the options they have, consider a coach. Sometimes, the team needs to escalate obstacles the team or servant leader has not been able to remove.
Work delays/overruns due to insufficiently refined product backlog items	Product owner and team workshop stories together. Create a definition of ready for the stories. Consider splitting stories to use smaller stories.
Defects	Consider the technical practices that work for the environment. Some possibilities are: pair work, collective product ownership, pervasive testing (test-driven and automated testing approaches) and a robust definition of done.
Work is not complete	Team defines definition of done for stories including acceptance criteria. Also add release criteria for projects.
Technical debt (degraded code quality)	Refactoring, agile modeling, pervasive testing, automated code quality analysis, definition of done