In response to several comments and conversations with people in the industry, several of us, most recently Chris Beale, Gary Baker, John Huston, and Daryl Kulak, have been converging on a definition of this new role, this new variety of leadership that we want agile teams to have. Many others are also involved in the conversation by now; too many to list here. The more we talk about it, the more comments we receive, the more all of us can see what we want. We are not in complete unanimity, by any stretch. Some folks think we are smoking crack. Not the first time. But most of us in the conversation can see useful patterns emerging.

This is a Big Role

We can also see that this is quite a lot of skill, experience, leadership, and sheer work to expect of one person. So part of my premise goes like this: the Agile Team Lead is the person that is ultimately accountable for ensuring that this work gets done. Depending on how the skillsets, preferences, team maturity, and other factors shake out for a leader and his/her team, the Agile Team Lead may well end up delegating much leadership to others (and other, subordinate leaders) within the team (the key phrase there being “within the team.”). So an Agile Team Lead might very well delegate to a Tech Lead, to an Agile Test Lead, to a Project Tracking Lead, to a Continuous Deployment Lead, etc., to cover off concerns. Whatever. The main point, of course, is that the buck stops with the Agile Team Lead. That person is, for whatever duration they are lead, the single wringable neck (for all responsibilities owned by the team itself).

This Role Must Emerge From Within the Team, and Must Satisfy All Stakeholders

So, while I am decidedly talking about DESIGNATING a person to act as the team lead, I am decidedly not talking about a traditional manager being imposed, permanently, from above, by fiat. There can be no Agile Team Lead designated FOR a team that does not know, admire, trust, respect, and like that person. Really, the final say in who leads the team should be the team’s.

This Role Should Likely Rotate

This may be something we want several team members to try out, to get good at. Team members could rotate through it, perhaps on a fixed schedule.

This Conversation, Um, Leads to a Few Others

I’d like to postpone to another series of blog posts, (already underway, however), the thorny issues around the enterprise structural implications that arise from this new role, and related observations. Suffice it to say, for now, that I personally don’t know how to make agile teams work, nor how to make this role of Agile Team Lead work, unless everyone in the team room reports directly to the Agile Team Lead, and no-one else. I mean that that one person is accountable for all of the incentive structure, performance evaluation, and (in large part) the hiring and firing for everyone on the team. This does not, again, mean that they own it entirely, nor that they do it permanently.

One other premise: we are deliberately attempting to erase old agile team management role and responsibility labels, like Scrum Master and Project Manager, which we increasingly believe to be fundamentally broken. We find the controversy around this topic to be perfectly matched to the stakes involved for the enterprise: our current patchwork of agile team management and project management and product ownership and process improvement management consists of more holes than cheese. Too many team-level concerns go unmet; too
many stakeholder concerns go unmet; too many user and market concerns go unmet. We have been in denial about this shortfall. Let’s give up that denial.

Love it or hate it, we need a paradigm shift around agile team leadership. We need that leadership to be better consolidated, better focused, better defined, and better matched to the dialectic between internal team needs and external stakeholder needs. We need one person who is held accountable the extent to which an agile team’s authority and responsibility are, in fact, congruent.

And we need none of that to violate the principles of self-organizing teams, leadership arising within the team, and teams owning the team’s concerns. The team owns all of these things, but the team lead is accountable for the team actually delivering it. Like I said, thorny.

Summary: What we Want in Our Agile Team Lead

Most of the following is in summary, bullet-point form, with a bit of explanatory text. We’re still mainly starting this conversation; much is left to be done to think this through from the perspective of the team itself, from the perspective of “continuous improvement,” from the perspective of individual Releases/Projects being executed well, from the perspective of Product Owners and Portfolio Owners, from the perspective of all of the different natural roles, responsibilities, skills, learning styles, and personality types that the team needs. These is just another draft sketch of the responsibility categories of this specific leadership role.

I suspect there are still several holes in our thinking, several bits of overlap, and perhaps some false premises. No matter. It’s out here now for us all to discuss. Better in front of your eyes here in public, than lurking, not yet “perfect,” on someone’s laptop.

Continuous Leadership

- Understand the goals of the organization, and the exact ways in which the team furthers them
- Provide a single point of leadership for the team; continuously wins the team’s implicit and explicit trust and admiration. Builds a “safe container” for the team within which work can be done creatively, collaboratively, honestly, accountably, and joyfully.
- Helps continuously build trust and respect between the team and its stakeholders (in a way that buys more and more freedom to speak up, freedom to experiment, freedom to “fail,” freedom to learn).
- Provide a single point of team accountability to the team’s stakeholders (in a way that ensures a single, coherent message to stakeholders about any team commitment about what went well, and what did not go well, how it happened, and why).
- Helps continuously improve internal team-cohesion: this includes mutual respect, mutual trust, morale, rapport, celebration, accountability, fun, and affection (yes, we have to like each other).

Continuous Planning

- Ensures that the team is making more and more accurate and precise commitments.
- Ensures that the team is delivering more and more predictably on its commitments.
- “Gets the lights on,” in every area of process and practice, and ensures everything is “big and visible,” both tactically within the team, and for purposes of publishing to external stakeholders.
- Identifies, gathers, and acts tactically on meaningful performance/progress metrics
- “Makes the plan the bad guy“: ensures that Definitions of Done, short-term scope, long-term scope, deadlines, quality measures, and all other commitments are expressed in “plans” that remove moral judgment and blame from continuous improvement. Plans include code, tests, cards, and other artifacts. The goal is to create an environment in which mistakes are expected, any shortfall is cheap and easy to learn from, and everyone’s reflex is to refine the “plan,” not to point fingers of blame.
- Drives Commitment-making for and with the team. This includes ensuring that the Definition of Done for any small or large commitment is precise enough and deterministic enough to reduce risk of misunderstanding, scope creep, and under-delivery as much as possible.
- Ensures that the plan changes with demand/supply.

**Continuous Execution**

- Monitors/manages team velocity/throughput. This includes work breakdown practices, estimation practices, tracking practices.
- Reports meaningful metrics/data within the team, tactically, and likely publishes different metrics/data outside the team to external stakeholders.
- Gets the team the resources it needs to deliver on its commitments; removes roadblocks
- Escalates blockages effectively. When the Agile Team Lead cannot unblock something, he or she escalates the issue to someone with the authority to remove that blockage, and makes clear to all stakeholders what the risks are of leaving that issue unblocked.
- Keeps flow, momentum and focus in the team.

**Continuous Risk Reduction**

- Continuously identifies short-term risks (both Sprint-scope and long-term, Release-scope and Portfolio-scope) across the full spectrum of people, process, technology, structure, and culture.
- Makes risks and potential impacts big and visible to the right people. This is another example of “Making the Plan the Bad Guy.”
- Ensures that risk reduction is integrated into planning and execution
- Measures the effectiveness of risk avoidance and the impact of unmitigated risk.

**Continuous Improvement (Agile Coaching)**

- Drives the improvement the overall Definition of Done (at each level of commitment scope, from the Story through the Release, and perhaps beyond). This has the effect of improving the quality of commitments, building a habit of overdelivering, building stakeholder trust, smoothing the flow of work (with fewer and fewer scope misunderstandings, production defects, and other “unplanned work phases”).
- Senses where performance breakdowns are occurring
- Creates big and visible means for measuring breakdowns
- Interprets data to identify biggest opportunities for improvement.
- Works with team to identify and implement improvements. This obviously includes owning a retrospective practice, and also includes owning the effectiveness and accountability culture around a reliable pattern of Inspection and Adaptation. It includes things like helping the team learn the difference between ego-based whining and team-based servant leadership, and the difference between vague negative generalization and concrete improvement proposal.
- Investigates and learns emerging practices from other teams, organizations, and communities.

OK, so once more, some disclaimer. This is work in progress. Expect more posts, other venues of conversation. Meanwhile, we welcome, indeed we desperately need, your input.

This entry was posted in Agility, Software Development, SW Mgmt by patrickwilsonwelsh. Bookmark the permalink.