The Rock Crusher and Work That Isn't On the Roadmap

contributed by Steve Adolph

"The hardest single part of building a software system is deciding precisely what to build. No other part of the conceptual work is as difficult as establishing the detailed technical requirements. No other part of the work so cripples the resulting system if done wrong."

No Silver Bullet: Essence and Accidents of Software Engineering. Computer
FP Brooks Jr - IEEE Computer Society, Washington, DC, 1987

Not All Work is Planned and Expected

The most dangerous four words in planning are "something has come up." This usually reflects work that will potentially affect, change, or even completely disrupt best intended plans, and it can't be ignored. Like all other work, this work needs to be made visible and managed. While the Rock Crusher is an excellent metaphor to help visualize and reason about how large Initiatives and features might flow through a backlog, on its own it can't represent all a team's work. We need to supplement it with guidelines for work that does not originate as large rocks.

Classifying Non Road Mapped Work

Many teams have both development and operational responsibilities, and the operational work tends to be time critical and arrives randomly. Many teams get themselves into trouble because they don't account for all their work. A common theme from many team retrospectives for why the team failed to meet their iteration goals is they failed to take into account all the work they were responsible for. This unaccounted for work is referred to as "dark work" – like the dark matter in the universe: it's difficult to see but its gravity has a significant effect on its surroundings.

We have tried to develop a strategy matrix for coping with non-roadmap work using two parameters:

- Do we expect the work; and
- Can we plan for that work.

Expected means we know that its coming – though we may not be certain of when. It's either on the roadmap, planned, or historical patterns indicate we can expect it. Unplanned work can disrupt our value flow. Unexpected means we only have, at best, a low probability assessment that the work will happen.

	EXPECTED	UNEXPECTED
PLANNED	Roadmap Work/Rock Crusher	Risk Management
UNPLANNED	Capacity Allocation	Retrospective

Regardless of whether work is planned, unplanned, expected or unexpected, the Backlog Owner is still accountable for the delivery of value. This encompasses all work and not just roadmap work.

Planned, Expected Work - Roadmap

Planned, expected work is classic roadmap work and is the Rock Crusher sweet spot. This is the work we are Crushing in our backlogs. This is the work we directly address in our Crush meetings. This is the work we are regularly using the Rock Crusher model to refine and re-prioritize with the goal of delivering the most valuable flow of value. We are aware of the work before we need to commit to it and can choose when we want to commit to the work (e.g., pull it into the iteration).

Unplanned, Expected Work – Capacity Allocation

Unplanned, expected is the work we know and expect to come to us, but we can't really plan for it like roadmap work because we simply don't know exactly when it will appear. We know it's coming because historically it always has, but we are not sure exactly when. Many teams aren't only responsible for the ongoing development of the system, but also configuring new users, security and server settings. These are the time critical defect fixes the team is responsible for and must fix. Then there are the transactional requests; the myriads of petty work requests that are sometimes too small to even name but seem to form a fog that slows all the planned work[1].

Unplanned, expected not only includes operational support but also looks up stream to analysis and design. Team members often take on Rock Crusher roles such as SME and even Analyst, and may have to take on Crusher work. If we're operating in a highly collaborative IT or product development environment, then team members will frequently be tapped for assistance for help during the Crushing processes. Talk to any senior team member and ask them how much of their time is required for these necessary activities.

While we can't know when a specific unplanned but expected request will arrive, we can use historical patterns to forecast how much of our development capacity we should allocate for those requests.

- For example, many teams just have a simple policy of not fully committing their capacity to backlog work. During iteration planning, a team may decide to simply hold back 25 percent and therefore only pull up to 75 percent of their measured velocity into an iteration.

There are two scenarios a team needs to watch for when using capacity allocation:

1. What happens when the unplanned, expected work exceeds the reserved capacity?
   In this situation, the unplanned, expected work threatens the team's ability to reach their committed objectives. There are two courses of action and both are at the discretion of the Backlog Owner because they affect the flow of value:
   a. Defer the unplanned, expected work because the Backlog Owner deems the planned, expected work more valuable than unplanned, expected work.
   b. Defer the planned, expected work because the Backlog Owner deems the unplanned, expected work more valuable.
2. What happens when the unplanned, expected work doesn't materialize or falls short of the reserved capacity?
   In this situation the team should simply "pull" more of their planned, expected work. While this additional work didn't form part of the team's planning commitment, there is nothing that prevents the team from pulling more Ready work into an iteration if they realize they can take on more work and it doesn't put their sprint commitment at risk.

Commitment based planning methodologies like Scrum may not work well for teams that have a very high percentage of planned, unexpected work – say well over 40 percent. These teams may prefer a pure Kanban system that is better suited to working with random arrivals of time critical work.

---

1   I apologize to all those who respect von Clausawitz, for appropriating and paraphrasing his famous "fog of war...." but this metaphor resonates so strongly that we believe it does apply here.

Planned, Unexpected Work – Risk Mitigation

Planned, unexpected work is the work that isn't in the backlog, nor is it part of the normal expected transactional work. Rather, there is some probability this work could suddenly appear and disrupt our flow. This is the domain of risk management where we plan for the unexpected by developing risk mitigation plans. Risk mitigation is the plan for how to break our plan and salvage some of our planned objectives.

A classic example of planned, unexpected work is when a new software release will occur during a team's planning time-box. Historically, the releases have gone well so there is no reason to expect disruption of the plan, but there have been times when things didn't go so well. Therefore, the team has a risk mitigation plan in place that can help them reach some of their objectives if there is unexpected work resulting from the release.

Unexpected, Unplanned Work - Retrospective

Unexpected, unplanned means we've been completely side swiped and there is no chance of reaching our iteration commitment. We can't defer the work, there is no risk mitigation plan and therefore there is no way to salvage the iteration. Service level agreements are now completely out the window. This situation is typical of an "all hands" defect fix. A mission critical system is down and everyone needs to fix it. A release was made to a client and they are completely dissatisfied.

Once the fires are out and the damage repaired, the team conducts a retrospective to discover what happened.

- Was this a black swan event that couldn't have been foreseen, or a marker of some systematic problem in our development process?
- If it was a black swan event, how was our response? Could we do better next time?
- If it is a symptom of a systematic problem, what can we do to fix it?

The team may try to salvage something out of the sprint. Scrum does have a concept of a sprint cancellation and sometimes this is necessary. Depending on the circumstances, it may be possible to replan the remaining time to get something of value done.

Under no circumstances should a team following an iterative methodology like Scrum, choose to either extend the iteration, or change the iteration calendar, in an attempt to reach the original iteration goals. This just sweeps the problem under the carpet. Agility is about making problems visible and dealing with them rather than ignoring them.

Summary

Much of the team's work will be transactional and not on the roadmap, and the Rock Crusher is not well suited for this kind of work.

The planned/unplanned, expected/unexpected model helps us manage this work:

Planned / Expected	Is classical Rock Crusher work.
Planned / Unexpected	Is the domain of Risk Mitigation.
Unplanned / Expected	Is historical transactional work the team regularly receives and is managed using capacity allocation.
Unplanned / Unexpected	Is a typically a black swan type of event and the only re-course the team has is to abandoned their commitment, deal with the event, and then retrospect to understand what happened.

Learn More
Rock Crusher for Backlog Management
Read:

1. Introduction
2. Ceremonies
3. Implementing
4. Roles
5. Crushes/Backlog Items

Considered Harmful
Anti-Patterns
Re-acquainting
Work not Road Mapped

View:

Rock Crusher Infograph

User Story Infograph