Rock Crusher Anti-Patterns

contributed by Steve Adolph

Rock Crusher anti-patterns are mostly all the classic backlog anti-patterns (absent Product Owner, no backlog refinement, too many Ready items, etc.), plus a few that the Rock Crusher may make more visible.

Backlog as a Reservoir – Unmanaged Inventory

This anti-pattern occurs when the backlog is a large queue of committed requirements the team must perform. The team may be doing incremental delivery of the solution (which is still a good thing), but they aren't agile. We frequently hear of teams having months of committed work in their backlog. These large backlogs of work tend to suppress learning and change because of the cost of change created by such a large inventory of work. The worst examples of this are when all the requirements from a Product Requirements Document have simply been dumped into the backlog. Evidence of this anti-pattern is when the team has months of Ready, committed work before they even start work.

The point of a backlog is to defer commitment and leverage learning to create superior economic value. Premature commitment to a set of requirements defeats this. It simply turns the backlog into a reservoir of pre-committed requirements. Flow is essential to creating economic value. In the Rock Crusher, flow means the rate at which new work is introduce (into the Rock Crusher) is balanced by the rate at which the team pulls work into the Thin Pipe, or Rocks are discarded through the Waste Gate.

Pushing work into the Rock Crusher faster than the rate at which it is being removed will cause Rocks to pool and result in wasteful inventory. The greater the inventory you are carrying, the less agile you will be. In any business, carrying some inventory is necessary, but it still must be managed.

Nothing is Ejected Through the Waste Gate

This anti-pattern is a classic symptom of the backlog as a reservoir anti-pattern. If nothing is being ejected through the Waste Gate, then all work entering the Rock Crusher is committed. There are situations where a very high percentage of work may be near ready, and that can occur when we are using the strategic-tactical or multi-horizon planning models for the backlog. In these situations, the team closest to the Thin Pipe may have a backlog of near ready and Ready Rocks to pull.

Indecisive Backlog Owner Or Backlog Owner Can't Say No

A potential root cause for the Nothing Ejected through the Waste Gate anti-pattern is the indecisive Backlog Owner or the Backlog Owner who can't say no. All work is important so therefore nothing is. The Backlog Owner owns the backlog and has the responsibility of prioritizing backlog items (Rocks) to deliver best value. The Backlog Owner doesn't have to be omniscient and intuitively know the value of each Rock. They can collaborate and work with the Solution Manager, Initiative Owners, SMEs and the team to determine the value of a Rock, calculate its size and determine if its a good bet. But, ultimately, it's the Backlog Owner who decides – the buck stops with them.

The Rock Crusher, like any model of the backlog, relies on a Backlog Owner who is able to clearly and decisively choose which Rocks are valuable, and which aren't. The Rock Crusher model visualizes that not all good ideas are going to make it to the Thin Pipe. Ultimately, the Backlog Owner must be fully capable and empowered to say "I am sorry but we aren't doing this."

Agility is Just A Software Thing

The attitude "agility is just a software thing" anti-pattern is another root cause of the backlog as reservoir anti-pattern. The Rock Crusher represents all the steps of the value stream to go from "concept to cash" and not just the software development steps. The classic Product Owner model assumed the value stream began and ended with the Product Owner. The Rock Crusher is a model for going beyond this myopic view of the world.

Many organizations, unfortunately, still believe agility is just a software thing and don't appreciated the importance of agility through their entire value stream. Evidence of this anti-pattern are organizations using Agile-Waterfall hybrid models like "water scrumming," where solution development still follows a Waterfall model with requisite flow impeding stage gates and the software steps follow some form of incremental development. This approach may help reduce technical risk for the software development steps, but it is like trying to resolve traffic flow through a city by widening two city blocks. After all the cost, expense and disruption, end to end travel time has not changed.

Absent Backlog Owner

This is the classic backlog management anti-pattern. This is the reason the Rock Crusher model splits the classical Product Owner role into the Backlog Owner role and six other roles (e.g., Customer, Stakeholder, Subject Matter Expert, Analyst, Initiative Owner and Solution Manager) to mitigate the problem of the absent Product Owner. Often, the individual tapped to be the Product Owner is an Initiative Owner, only interested in their initiative, or a Solution Manager who doesn't care about "nerd details."

The Rock Crusher's approach to backlog management roles offers a more realistic approach to roles and better aligns interests with their participation in managing the backlog. However, as Alistair Cockburn once remarked, "people trump process." Nominating an individual who is uninterested in serving as the Backlog Owner versus being an Analyst, or Solution Manager, or Initiative Owner, will still impede flow. The Backlog Owner is a critical role, and an individual shouldn't be nominated to the role just because they're available.

Unsupported Backlog Owner

This is an anti-pattern that Rock Crusher can explicitly call out the unsupported Backlog Owner. The Rock Crusher mantra is "it takes a village." The Rock Crusher makes all the work to go from "concept to cash" visible, and there are numerous roles that must collaborate to get Rocks Ready. Only in the smallest close-knit organizations can one individual do it all – classical Product Owner.

Initiative Owner and/or Solution Manager Overrule Backlog Owner

The Backlog Owner owns the backlog – full stop. While the Initiative Owner and Solution Manager are accountable for their respective initiatives and solutions, they can't overrule the decisions of the Backlog Owner. This creates confusion and conflict within the team. There may be several reasons for this:

- The Backlog Owner is not "stepping up to the plate" and getting their job done so either the Initiative Owner or Solution Manager step up to fill the void.
- Corporate compensation and reward mechanisms tie compensation to getting initiatives done and solutions delivered. This sets up conflict between the Backlog Owner and Initiative Owners and Solution Managers.
- Lack of clarity in accountability between the Initiative Owner, Solution Manager and Backlog Owner.
- Backlog Owner is a direct report to the Solution Owner or Initiative Owner and they are using their positional authority to overrule the Backlog Owner – showing a lack of trust.
- Lack of clear goals means Initiative Owner, Solution Manager and Backlog Owner have conflicting goals.

Backlog Owner Ignores Solution Manager and/or Initiative Owner

While the Backlog Owner has final authority over the prioritization of Rocks in the Rock Crusher, the role is a collaborative role. The intent of the Backlog Owner is they collaborate with all other Rock Crusher roles to maximize the value delivered through the Rock Crusher.

- For example, this could lead to situations where the Backlog Owner is forced to tell an Initiative Owner their initiative won't have access to the team and is a candidate for the Waste Gate. This should be an explicit conversation, not just letting the initiative languish.

Non-Backlog Work is Unmanaged

Not all work is on the Road Map, expected and planned. For many teams, a fair percentage of work is transactional, expected-unplanned, or even unexpected-planned. Frequently, this type of work is not managed; the result is the team is overwhelmed and not operating a sustainable flow. Evidence of this anti-pattern is teams frequently referring to their day job or other work. Further evidence is the team not being able to reach their commitments because they keep "getting interrupted" by the shoulder taps of unmanaged work. The Rock Crusher uses both capacity allocation and risk mitigation to manage this kind of work.

Team has Multiple Backlogs

The Agile Mantra is "bring the work to the team," therefore a team has only one backlog. A classic anti-pattern is the team has multiple backlogs, often one for each "project" they're working on. The Rock Crusher attempts to mitigate this problem by separating initiatives and Initiative Owners from Backlog Ownership. All of a team's work is in their backlog. Backlog management becomes - at best - tricky if a team has to contend with multiple backlogs. A variant of this pattern is when team members are on multiple teams. This dramatically increases coordination overhead, increases delays and impedes flow.

Infrequent Crush Meeting

Crush meetings are the Rock Crusher's heartbeat; not holding Crush meetings means we're not managing the flow of work through the Rock Crusher. The team will then be starved for work or working on the wrong work (e.g., less valuable). Usually, a symptom of this anti-pattern are long, confused iteration planning meetings where the team now has to "Crush in a panic" and "Crush with ambiguity" to pull work for their iteration.

Too Many Crush Meetings

Frequent Crush meetings start to take away from getting the job done. But, more seriously, if there are excessive/frequent Crush meetings, team members may disengage from them. There are several reasons for too many Crush meetings:

- A "command and control" Backlog Owner who is using the Crush meetings to control the work done by the team.
- A team is uncomfortable with ambiguity and unwilling to take the initiative and develop work without directives. This may be evidence of team members wanting a paper trail to defend their choices.
- Team is unwilling to pull work through the Thin Pipe (e.g., code) unless Rocks are rigorously specified.
- A team who is only interested in one part of the work (e.g., coding) and the Crush meeting is the only way to get their input on work.
- Team is completely unfamiliar with the problem domain.

There is one situation where frequent Crush meetings are temporarily appropriate. This may be early in a new team's formation with a new green field solution, and there may be a legitimate necessity to have frequent Crush meetings. However, this can quickly devolve into a way of avoiding taking the initiative.

Lack of a Strategic Crush Meeting

A small team can get away with one level of Crush meeting. A small team is something like a classic Scrum team (7+/- 2 people) with a Backlog Owner, perhaps a Solution Manager or an Initiative Owner, and an Analyst. This model covers a large percentage of work. However, start adding multiple Initiative Owners, or multiple teams, then more coordination is required to manage the Rock Crusher and we need to have a multi-level planning model. For the Rock Crusher, this is done by splitting the Crush meeting into tactical and strategic meetings. The lack of strategic meeting means important strategic prioritization decisions are either made during the tactical meeting or not made at all. It also runs the risk that a significant portion of the audience is not interested in the conversations taking place and will start avoiding the meeting because the agenda is of no interest to them.

- For example, it may be the Initiative Owners and Solution Managers aren’t interested in more tactical “nerd details” and developers may only have a casual interest strategic “portfolio” issues.

No Definition of Ready (DoR)

There is no Definition of Ready, which means we don’t have any guide informing us what a Ready Rock looks like that can be pulled through the Thin Pipe. A Rock that is too large can either break or plug the pipe - that requires so much remedial work to get Ready - that we can’t get a predictable flow of value through the Thin Pipe. On the other hand, Rocks that are “too small” (that is, too refined) simply wasted time and effort, the classic Lean waste of “over-processing” doing more work than is necessary.

Rock Crusher is Invisible – No Kanban States

The whole point of the Rock Crusher is to make visible all work to go from concept to cash. If there is not an easily accessible visual representation of the Rock Crusher, e.g., some kind of Big Visible Information Radiator (BVIR), then all of the Rock Crusher is for naught.

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