

WORK IN PROGRESS

Idea in brief

In software development it's easy to ignore a big pile of partially done work sitting in a queue because software is intangible and invisible. Requirement specifications, an amount of code (working or not), or even a complete, finished product which has not yet gone live aren't sitting in a big pile on a desk – they're tucked away on a server somewhere. This work in progress (WIP) represents an investment which is not yet creating any value.

Controlling queues can lead to faster delivery and one of the best ways to manage them is to constrain the amount of work in progress at any one time.

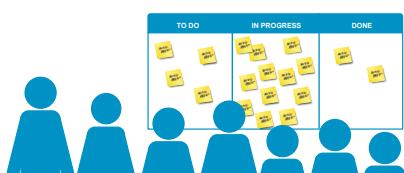
In order to manage our queues and WIP we need to make them visible. A good visualisation of the team's work means that you can cut down on progress reports and updates. This makes it easier to spot emerging queues whether for individuals, processes or departments, allowing the team to revise estimates and consider appropriate action.

In a self-managed visual system, team members (or teams) only 'pull' the quantity of work they can deliver. This is, in essence, a limit to the WIP in the system. Agile methods depend on this pull approach.

The result of limiting the amount of activities being worked on at any one time means that the work we have committed to will get done faster. Rather than spreading ourselves between three or four projects, we commit to only one and that project as a whole is finished more quickly.

Ideas in practice

We create a to-do list for the team to make WIP visible



- Make WIP visible for the team.
- Visibility provided by the board allows for the management of WIP and queues.
- The board should be dynamic, preferably physical and easy to understand at a glance.
- Update it daily (at a minimum!).
- Be wary of having more than one place to update the information.

We can use WIP limits to cap any one person or process



- Introducing a WIP limit is a simple as capping how many tasks one person or process can have in progress at any one time.
- WIP limits lead to feedback – observe and adjust as necessary.

"Do not fret over a decision to establish a WIP limit. Choose something!"
David Anderson

VISUALISATION

helps facilitate a 'pull system'

- Rather than having work assigned by a manager, the team takes responsibility for pulling work from the board.
- Without explicitly restricting the amount of WIP, individuals are able to judge just how much work they take on.
- The real benefit of this is to stop individuals from becoming overloaded.

Delaying starting on something can be a cheaper place to queue software than delaying it in-flight



- In holding work in a queue we are attempting to improve the flow.
- Different queues have different costs – it is often cheaper to delay starting on a piece of work than to queue it in-flight.
- In Agile teams – this held work is often called the backlog.
- Pull work from the backlog as capacity becomes available.

What?
Reject work?
Are you serious?

- Blocking demand or rejecting work lowers WIP and helps prevent the organisation from becoming overloaded.
- By focusing on fewer things, individually they are each more likely to get finished quicker.
- It's not the number of things currently being worked on that matters... it's the value that is delivered.

Alternative WIP controls

1. **Throttling** – raise the criteria for starting a piece of work.
2. **Purging** – stop low value projects to make way for new ones.
3. **Shedding Scope** – limiting our input.
4. **Trimming** – reducing the scope of an already in-flight feature.
5. **Timing** – to manage an in-demand resource.