Prioritizing Toil

Site Reliability Engineering

Objectives

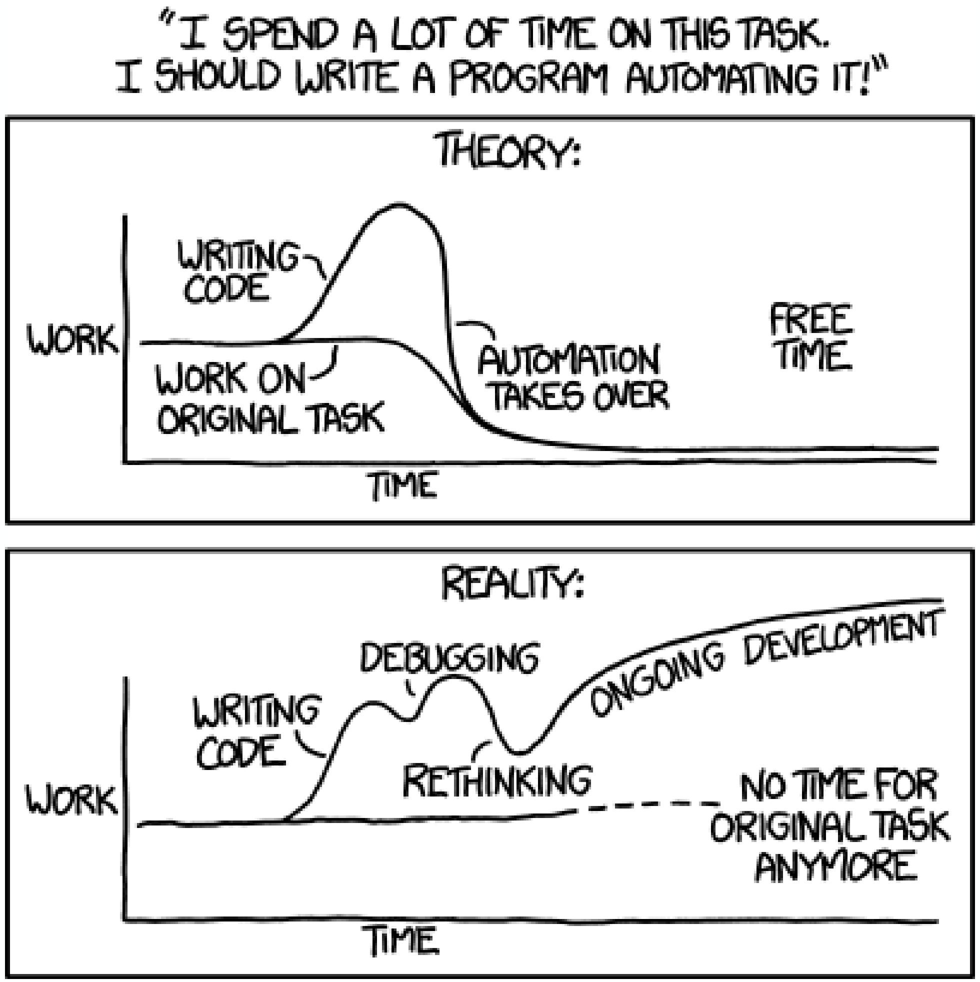
In this module we will look at how to determine the order to work on toil automation.

Learning Objectives

- Create a toil backlog
- Prioritize the toil
- Prioritize over the error budget
- Prioritize the pipeline

{mthree}

Time Spent on Toil Estimation



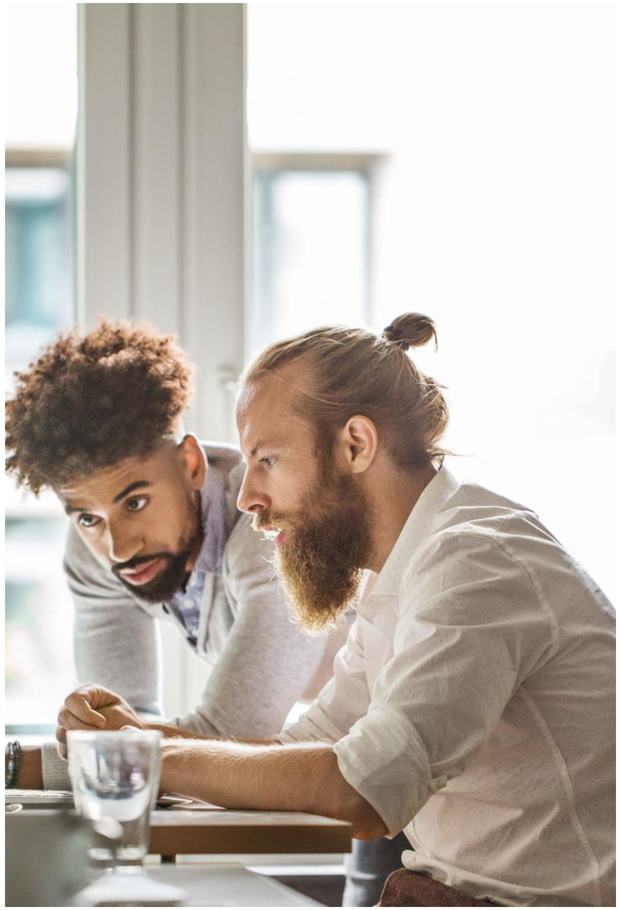
Source: XKCD – Automation

{mthree}

Toil Backlog

Keep on top of toil

- ↘ Aim to reduce the backlog as part of the development cycle
- ↘ Manage it like you manage your application development
  - >>> Backlog
  - >>> Task sizes
  - >>> Sprints
- ↘ Toil should be added to the backlog when it occurs
- ↘ Backlog categorization
  - >>> Manual – needs documentation
  - >>> Automated – provided toil calculation



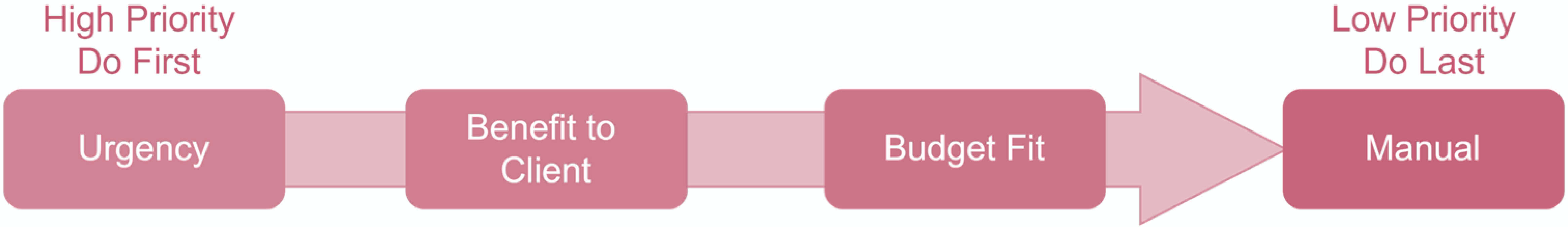
{mthree}

Weighting Toil

- Identify the toil type
- Record timings
  - Actual hands on
  - Number of checks/views before task completes
  - Total run time of toil (includes non-hands on)
- Record enjoyment level
  - Easy toil
  - Moderate toil – "it's ok, just not too often"
  - Draining – too much thought required, or too many places to investigate
- Automation capability
  - Can it be automated? (yes/no)

{mthree}

Prioritizing



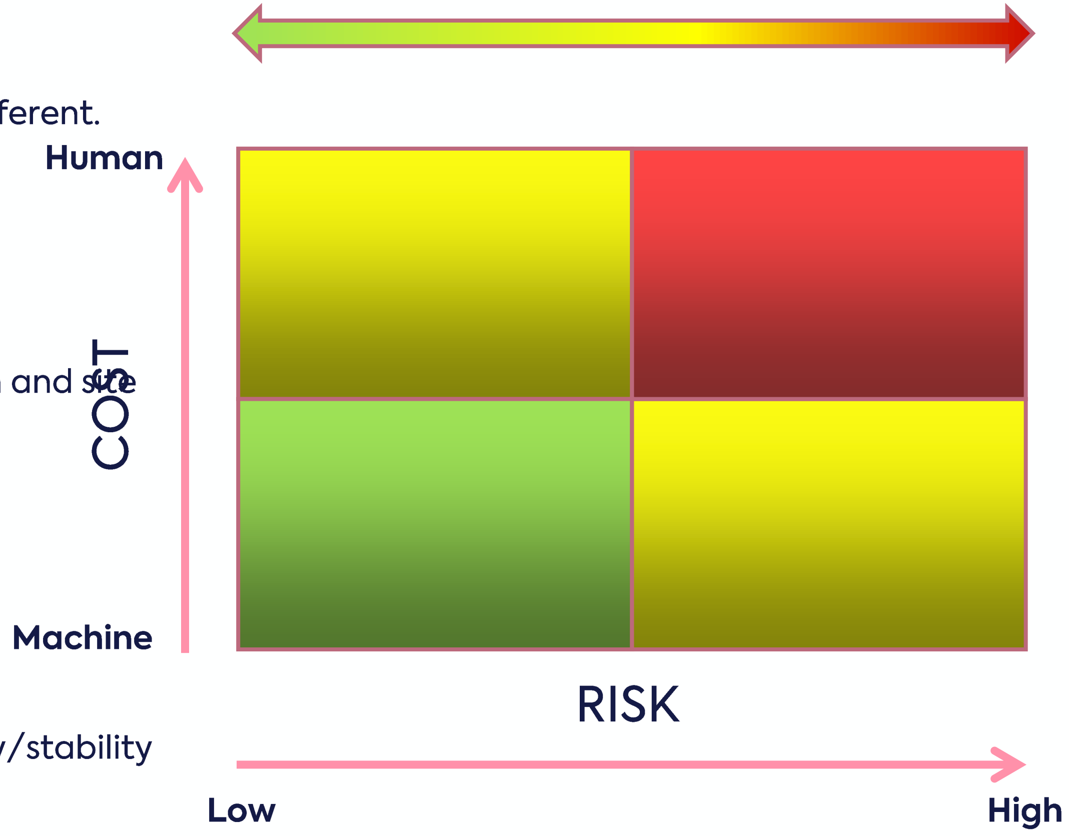
{mthree}

TOIL is repetitive work that has cost and/or risk that is significant to the **site**
Has no enduring value
Has well defined processes for automation
Can be eliminated through lack of use or automation

Cost of elimination > cost/risk performing   Cost of elimination < cost/risk performing

KEY:
Identify your indicators, each site is different.
How often, and grouping for KPIs
- Alerts with manual actions
- Tickets from Incidents or Problems
- Site checks BOD and EOD, etc
- Constantly failing processes
- Other interruptions **site** related

TOIL must be meaningful to your team and site



KEY:
TOIL elimination measurements
- Reduced MTTR
- Ticket/Alert reduction
- > focus on automation and reliability/stability
- Increased team moral

KEY:
TOIL measurement through Jira/SNow, etc
- Use tagging on tickets/jobs
- Surveys to spot low moral through toil
  - To determine if it is real toil

KEY:
TOIL reduction investment to consider COST and RISK dimentions

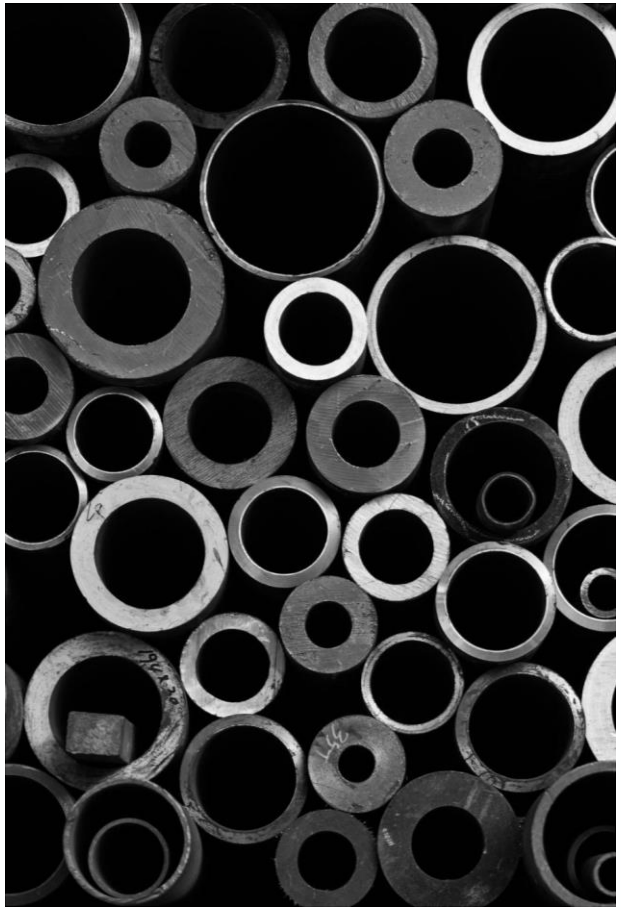
KEY:
Automation vs Elimination = key to investment

KEY:
If anyone can do it then do we need to eliminate?

OVERHEAD is repetitive work not related to the **site**, and will vary like toil across sites != **TOIL**

Prioritize the Pipeline

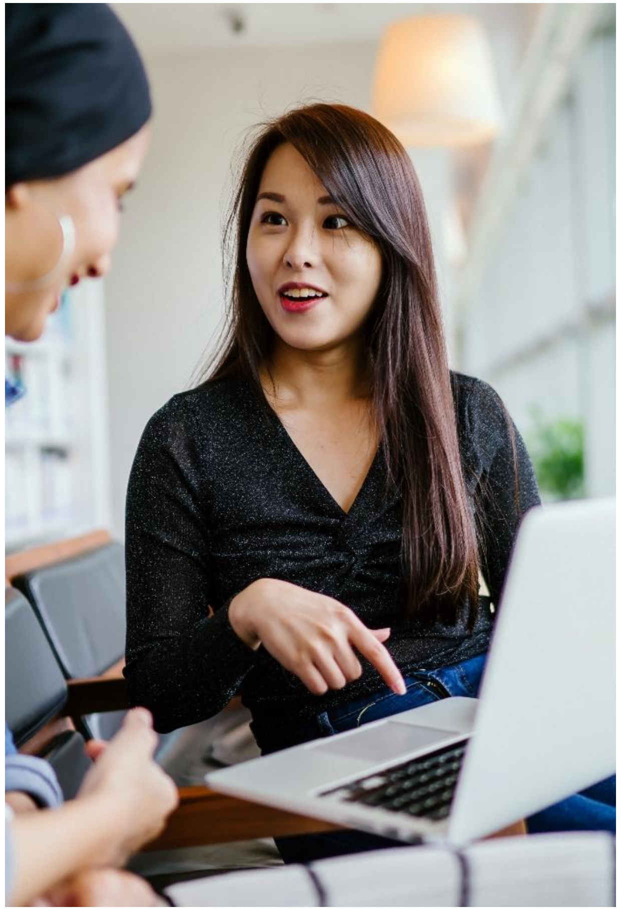
- Toil tasks may also be present in your pipeline
- Pipeline is as important as production releases
- Production should always be priority
- But progressing the application is also important
  - >>> If the pipeline breaks or testing isn't caught properly what happens?
- We don't want toil building up in the pipeline



{mthree}

Pipeline Toil Prioritizing

- Manual steps should be reduced
  - Typical automation should be
    - Detecting changes in source control management (SCM) systems
    - Any testing
    - Package promotion
- Manual steps that could be reduced
  - Advanced automation
    - Automated sign off
    - Automated deployments
  - These require good APIs and good testing



{mthree}

What is the toil?

Spreadsheet Word Docs No method to record work	Git Update orderbook Git Update Infra	Currently no testing! Dev performs manually	Push Git repo to infra PR to merge	Check all links work in Dev and app responds OK	Promote container image Update Infra Git repo PR	Bug reporting Incident response Post-mortem
Prioritze reported bugs	Development	Build/run unit test	Deploy new code to pre-production	Perform QA testing	Deploy new code to production	Monitor code
Product manager	Software engineer	Software engineer	Operations engineer	QA specialist	Operations engineer	Operations & Software engineer

Toil values

LT: 3 – 5 hours	LT: 10 minutes	LT: 60 minutes	LT: 10 – 30 minutes	LT: 10 minutes	LT: 60 minutes	LT: 1 – 3 hours

Summary Q&A

{mthree}