Applying Data Analysis to Software Development

Laurence Chen | CEO @ REPLWARE
https://replware.dev

About Me

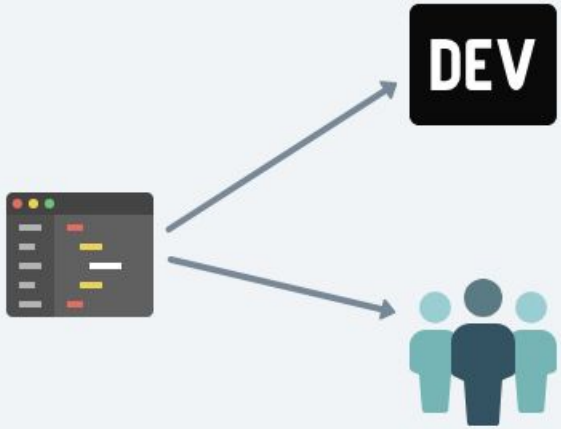
- Clojure community organizer
- IT Consultant, speaker, writer
- https://leanpub.com/errors_to_innovation/ | 從錯誤到創新

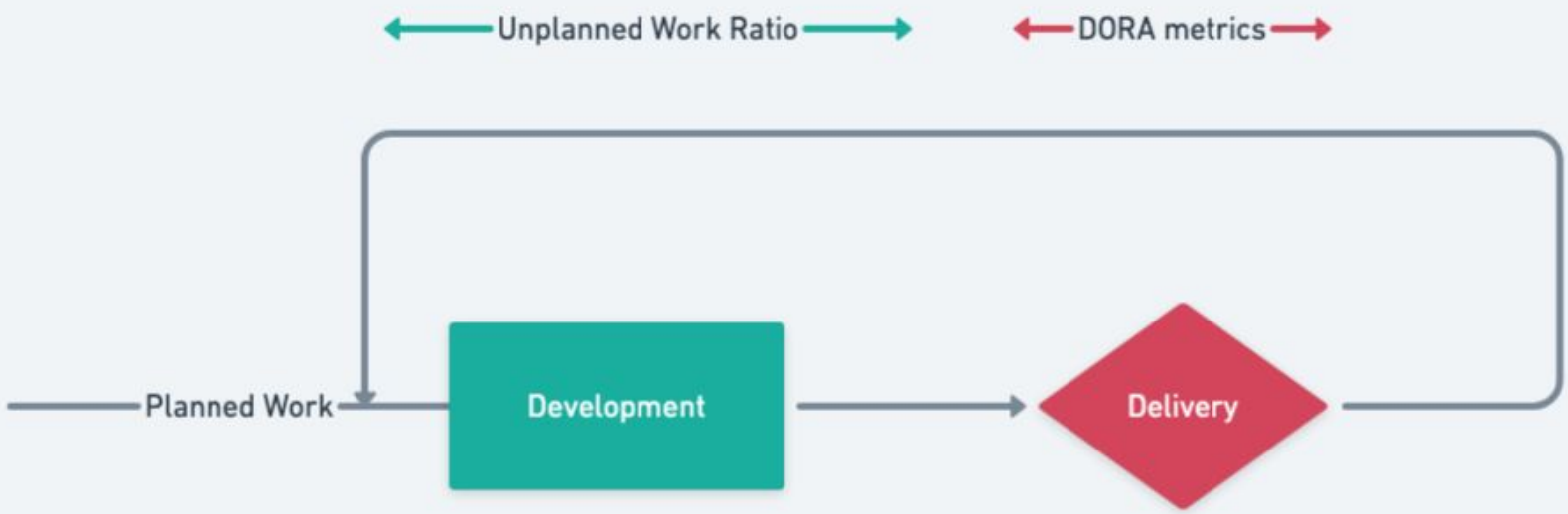
Agenda

- Why do you care?
- DORA metrics
- Unplanned Work Ratio
- Idea Flow Ratio

Why do you care?

- We, developers, want to refactor.
- Management people want features.





- Typical Activities
  - Understanding existing code
  - Change the code
  - Validation
  - Debug
  - Rework

- Typical Activities
  - Operational practices
  - Deployment
  - Monitoring


如何在職場展現你的成熟度
LIGHTYEAR
用「OK、好」取代[redacted]娘

DORA metrics - DevOps KPI

- Deployment Frequency
- Lead Time for changes: The amount of time it takes a commit to get into production
- Change Failure Rate
- Mean Time to Restore

DORA metrics - Benchmark

	Elite	High	Medium	Low
Deployment Frequency	on-demand	once per week ~ month	once per month ~ 6 months	< (once per 6 months)
Lead Time for changes	< 1 hour	1 day~ 1 week	1 month ~ 6 month	more than 6 months
Change Failure Rate	< 1 hour	< 1 day	1 day ~ 1 week	more than 6 months
Mean Time to Restore	0%~15%	16%~30%	16%~30%	16%~30%

How to improve DORA metrics

- Automation of testing and CI/CD
- Break the changes down into small iterations.
- Observability in production environment.

Discussion of DORA metrics

- Measure the global outcome.

  => Every metric is directly related to **the release**.

- Goodhart's Law: "When a measure becomes a target, it ceases to be a good measure."

  => Paired Metrics

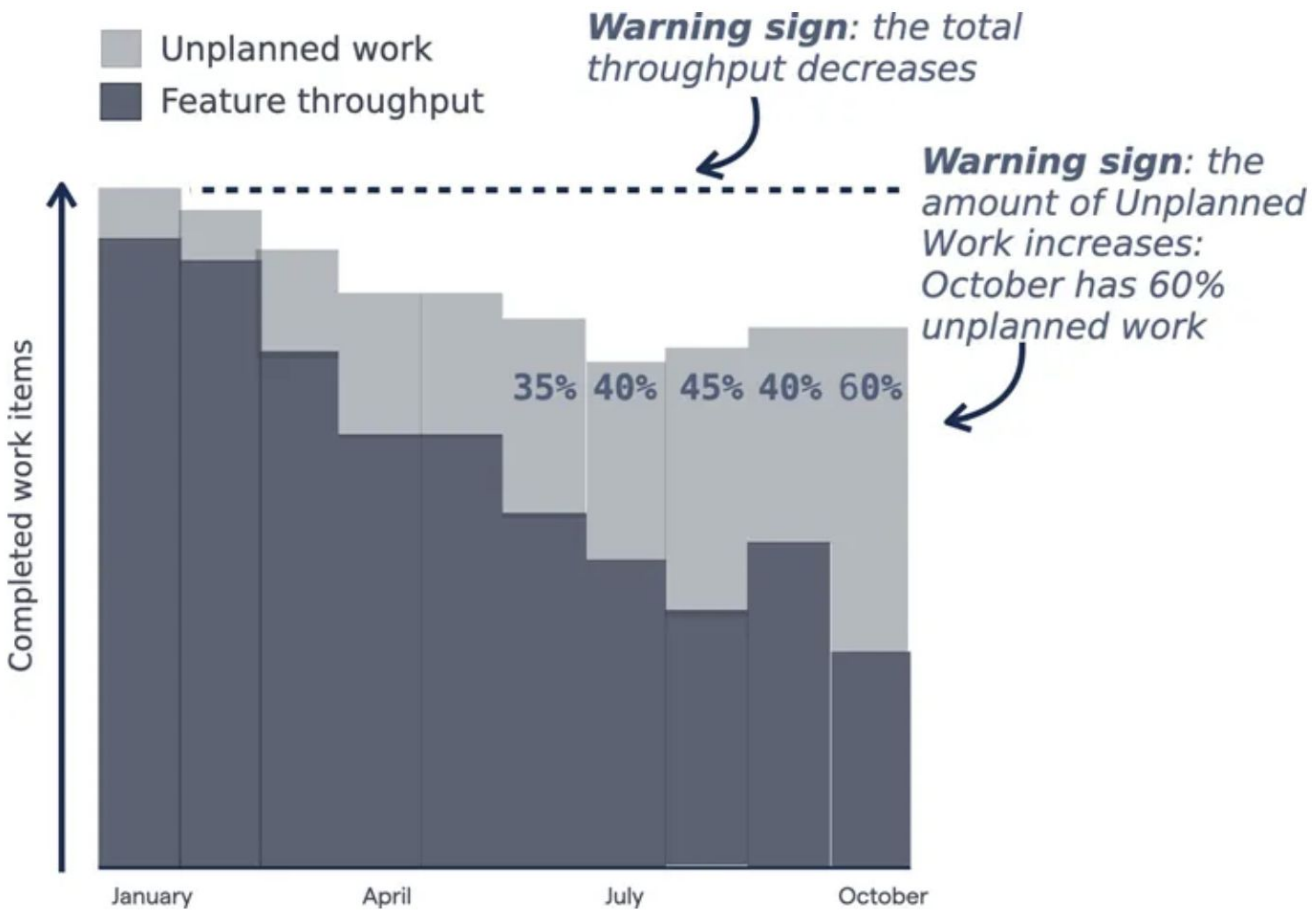
  => Two for **velocity**, and two for **stability**.

- But, how about the tasks happening before **the commits**?

Unplanned Work Ratio

- Unplanned work is anything you didn’t anticipate or plan for, such as bug fixes, service interruptions, or flawed software designs causing excess rework.

- If it is more than **15%**, then you have troubles.


Unplanned work
Feature throughput
Warning sign: the total throughput decreases
Warning sign: the amount of Unplanned Work increases: October has 60% unplanned work
35%
40%
45%
40%
60%
Completed work items
January
April
July
October

Unplanned Work Ratio Benchmark

- Highest performers: < **5%** unplanned work
- Lowest performers: > **50%** unplanned work

How to improve unplanned work ratio?

- Your code? But where?
- How about our work flow? How to measure it?

Code Scene - Where to refactor in your code?

- Your Code as a Crime Scene - by Adam Tornhill

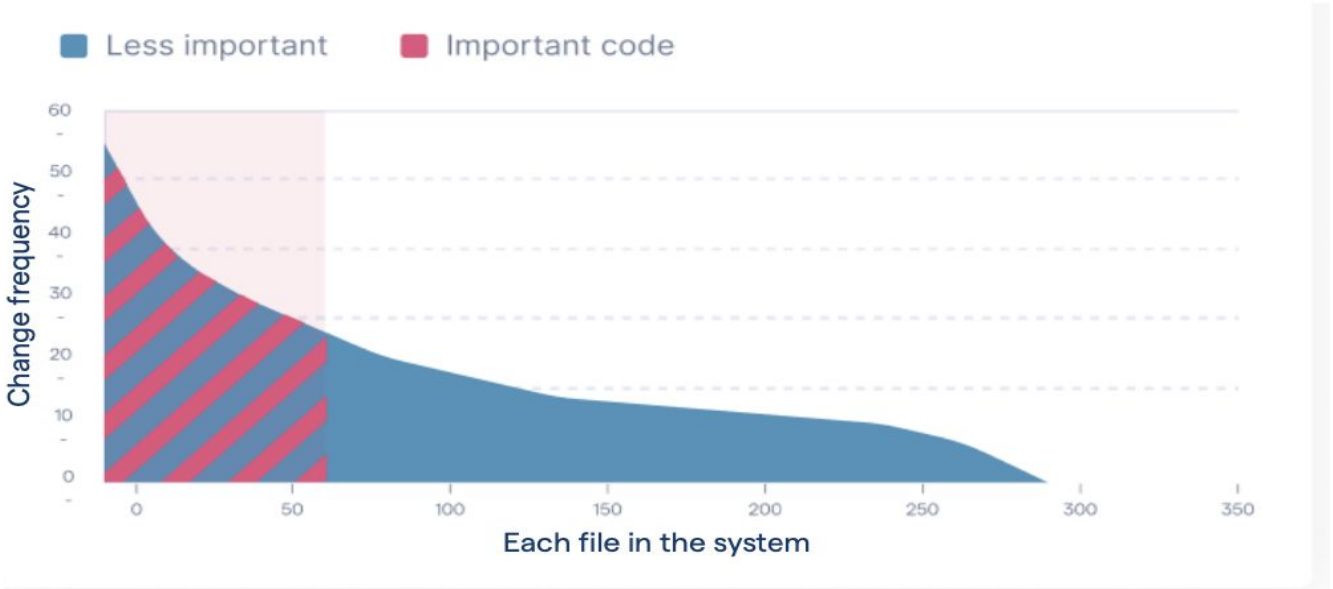


Figure 2. Change frequency of source code files

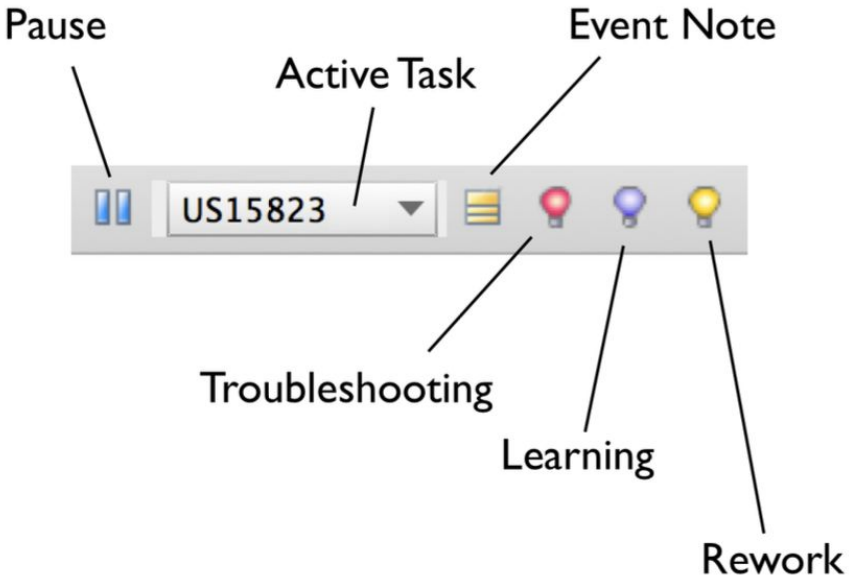
Idea flow

- Idea Flow - How to Measure the PAIN in Software Development - by Janelle Arty Starr

- Optimize **the flow** instead of optimizing **the code**.

How to measure the flow?

- Editor Plugin

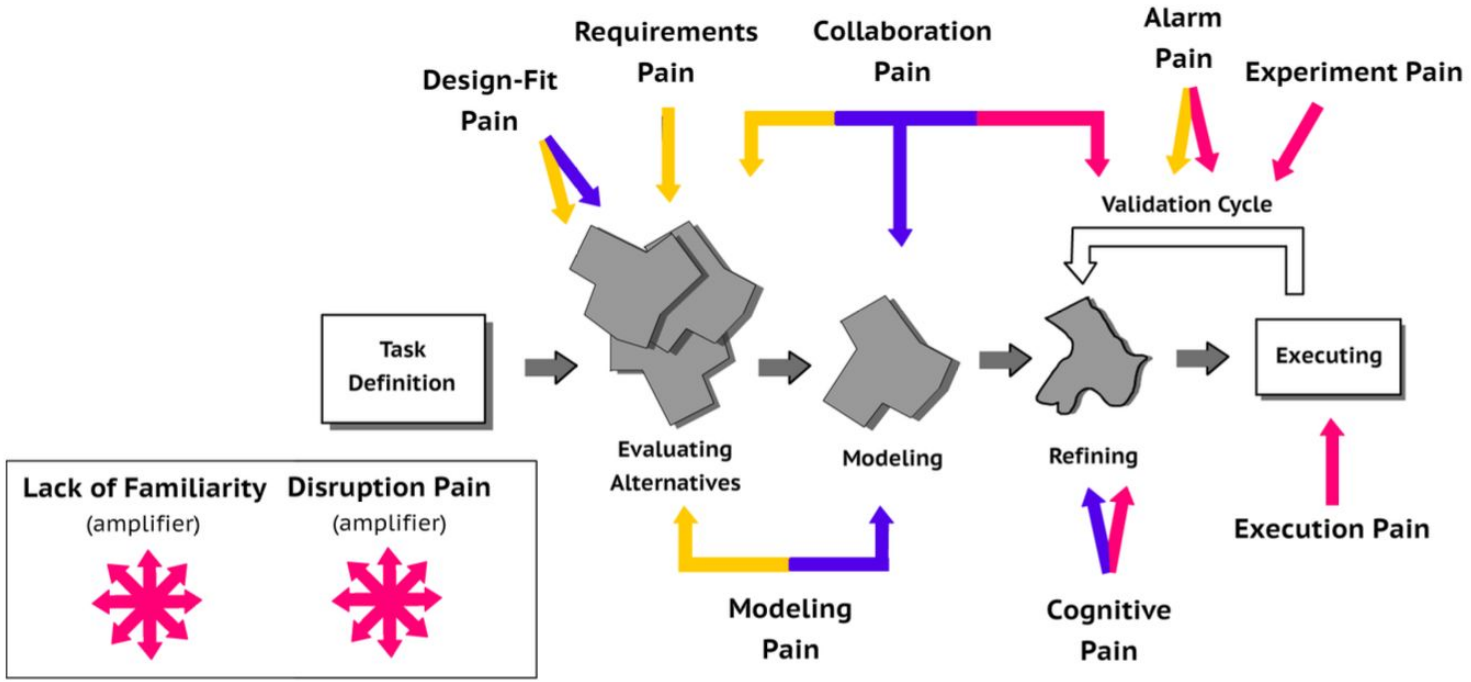


Focus on improving the idea flow

- Before

- After

The Ten Pains of Software Development



Four Most Important Pains

- Modeling Pain (Task Complexity)
  - When it's difficult to build a conceptual model of how the software works because the code is difficult to scan.
- Experiment Pain (Task Complexity)
  - When it's difficult to setup experiments, run experiments, or figure out what's going on, it's experiment pain.
- Lack of Familiarity Pain
  - Familiarity changes the perceived complexity of the task.
  - Don't be fooled.
- Disruption Pain

In Conclusion

- Do high ROI (return on investment) task.
- Talk the walk and walk the talk.
- Visibility and Control

Q & A