MICHIGAN IT

Agile Software Engineering With #NoEstimates

Business wants to know:

- **How many features can you get done?**
- **When will features be in production?**

Story estimates provide unreliable answers to these questions. Further, the process of estimating stories can be harmful.

Even perfectly accurate estimates do not change the value of a story. Repeatedly deciding to defer very valuable but expensive stories in favor of cheap, nice-to-have stories will not instill a sense of trust in the customer. **Knowing the costs of user stories distracts management from remaining focused on meeting the most important needs as quickly as possible.** On top of that, **estimates are always wrong.** Sometimes wildly so. Why make bad decisions based on worse data?

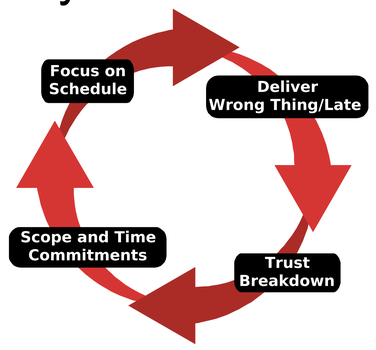
It turns out that **deliberately avoiding estimating helps teams focus on delivering the most valuable business needs**, in a more efficient, incremental way. Progress **transparency can be provided with story counts and cycle/lead times** to show wait times in the queue/backlog without having to estimate anything. These metrics are simple and empirical. By writing like-sized and independent stories, teams can provide much of the visibility that estimates are supposed to deliver.

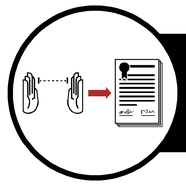
#NoEstimates focuses on the following questions:

- **What does the business need most to be delivered?**
- **What user story (feature or bug fix) delivers the most business value?**
- **What is the smallest, most incremental way that value can be delivered?**
- **Can that user story be split into even smaller pieces that still each deliver value?**

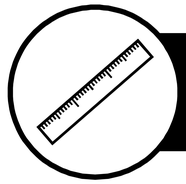
Estimates

Cycle of Mistrust

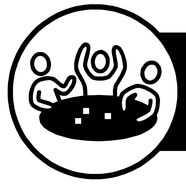




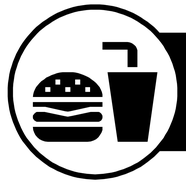
Estimates become commitments of both scope and time.



Measuring/comparing team performance with story points.



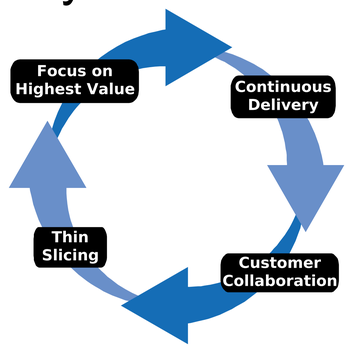
Encourage game playing and dishonesty.

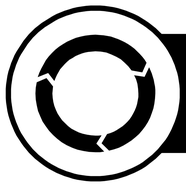


Distort prioritization based on cost instead of value.

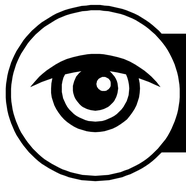
#NoEstimates

Cycle of Trust

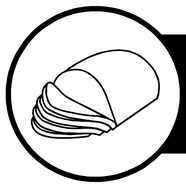




Deliver value continuously.



Focus on doing the most valuable thing next.



Slice as thinly as possible while still delivering value.

Provide transparency with simple, empirical metrics.

MEDICAL SCHOOL INFORMATION SERVICES
UNIVERSITY OF MICHIGAN

Evolution of Metrics

Traditional Metrics:

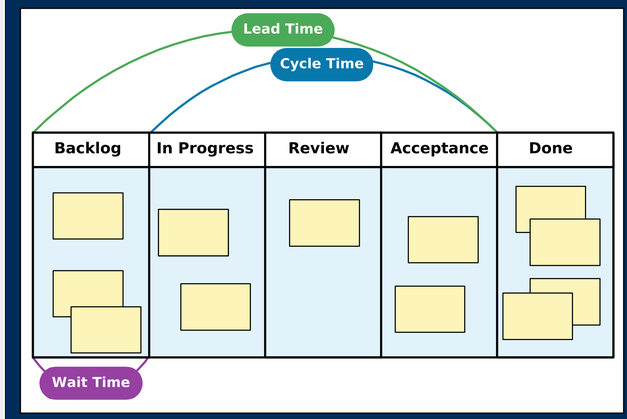
- Lines of Code
- ROI
- Estimates vs. Actuals in Hours
- Utilization

Agile Metrics:

- Test Coverage
- Velocity
- Estimates vs. Actuals in Story Points

#NoEstimates Metrics:

- Story Counting
- Lead Time
  = Average Time From Backlog to Done
- Cycle Time
  = Average Time From Started to Done
- Wait/Queue Time
  = Lead Time - Cycle Time



Created by Steve Coffman, Bryan Martyn, and Andy Vella

Clipart via The Noun Project

Send feedback and ideas to:
agile.me.this@umich.edu