RALLY SOFTWARE

AGILE MEASUREMENT

OPPORTUNITY

BURNS AWAY OVER TIME

BUCKET BRIGADE GAME

Which is faster?

- Bucket line
- Bucket runner

WHAT DID WE LEARN?

Our intuition is often wrong.

Measuring is frequently the only way to decide on the best alternative and change beliefs.

Coordination matters.

BUCKET BRIGADE
ROUND 2

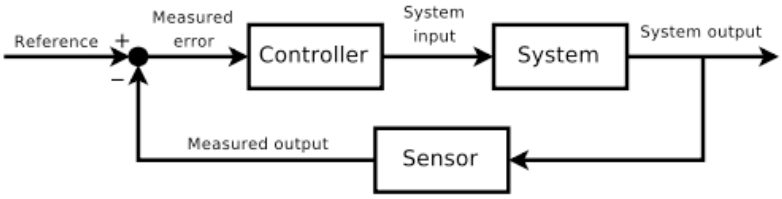
WHAT NEW DID WE LEARN?

Context matters.

Only use industry studies as a starting point.

DO YOU OWN MEASUREMENT.

AGILE IS…



A set of feedback loops with appropriate emphasis and frequency

Appropriate valuing of tacit knowledge and verbal information transfer

Cultural values

FEEDBACK EMPHASIS

	TRADITIONAL	AGILE
SWEET SPOT	MANUFACTURING AND SERVICES	ENGINEERING AND DEVELOPMENT
PROCESS FEEDBACK	**FREQUENT, FORMAL, AND CAN BE HEAVY WEIGHT**	LIGHT BUT OFTEN
PRODUCT AND DESIGN FEEDBACK	**?**	**EARLY, OFTEN, AND CLOSE TO THE CUSTOMER**
PLAN FEEDBACK	LONGER-TERM	FRACTILE
CULTURE	STRUCTURED	COLLABORATIVE

BASIC ASSUMPTION

Better insight → Better decisions → Better outcomes

How to get better insight?

PLAN-DRIVEN	AGILE
Documented knowledge	Tacit knowledge
Quantitative insight	Qualitative insight
Organization	Team
Advantages: • Precision • Counter folk lore or overcome faulty intuition • Motivate us to do what we should, over what we want	Advantages: • Lower costs • Greater agility • Takes particular situation into account

IMPLICATIONS FOR **AGILE MEASUREMENT**

1. Product measures matter more
2. Must compliment (rather than attempt to replace) tacit knowledge and verbal information transfer so visualization and interaction is critical
3. Team-level measurement more important
4. Passive data collection because culture will not tolerate more

5 "STANDARD" OUTCOME MEASURES

1. Cycle time for features (user stories)
2. Cycle time for defects
3. Productivity =
   size / (schedule * man-hours)
4. Defect density
5. Schedule deviation (predictability)

Trending as important as level

FROM WHERE?

The data passively gathered in ALM tools like Rally can be leveraged to calculate these measures.

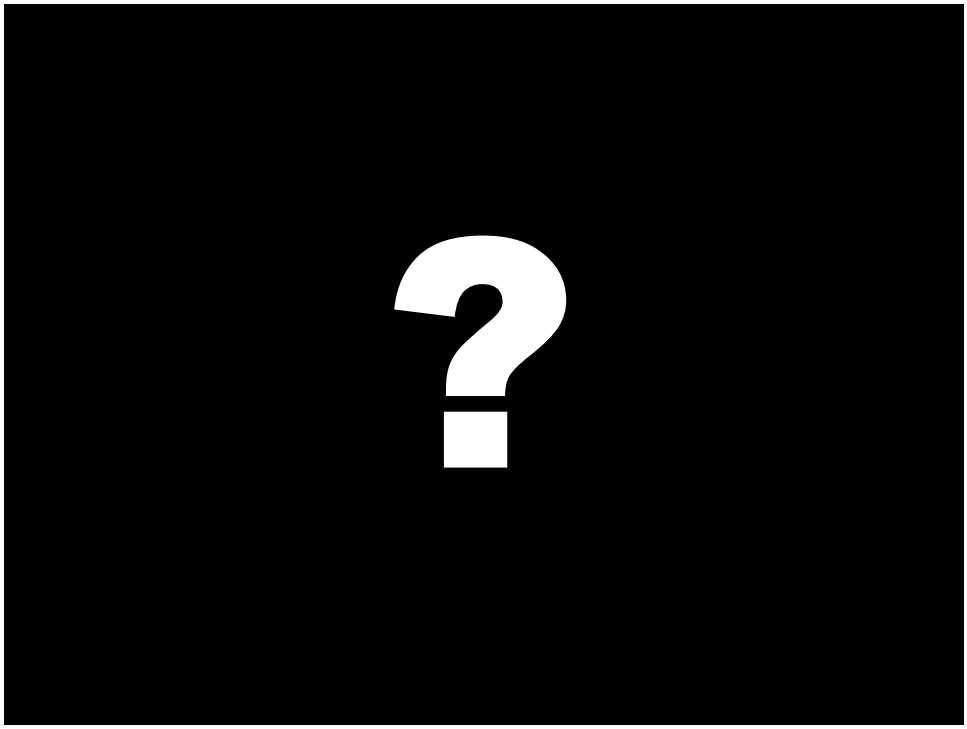
Nuance depends upon:

- Task breakdown and measured in hours?
- Hours remaining updated daily mid-sprint?
- Actual hours for tasks recorded after-the-fact? (or some other form of estimated versus actual)
- Integrations:
  - Defect data traceable?
  - Source code repository linked?
  - Build data available?

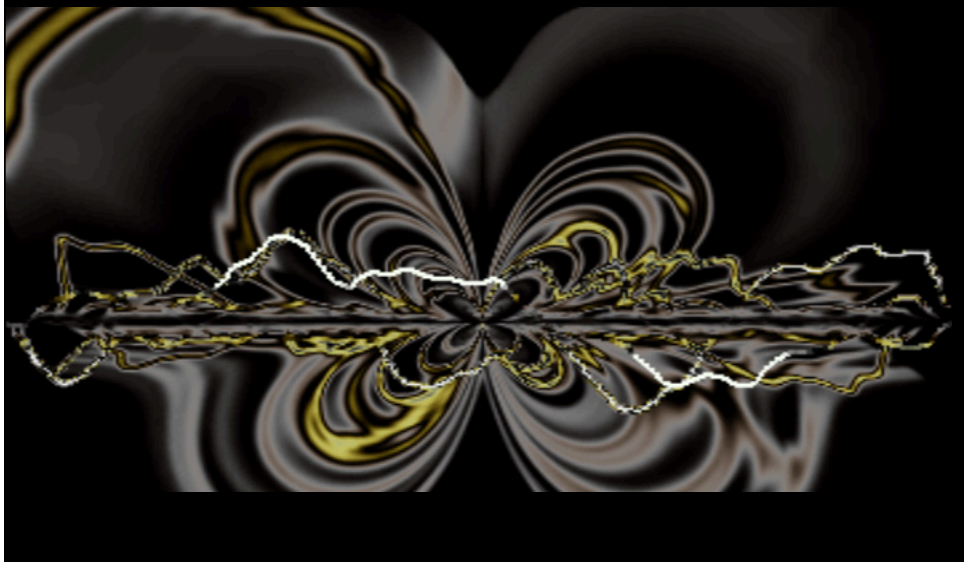
18

21

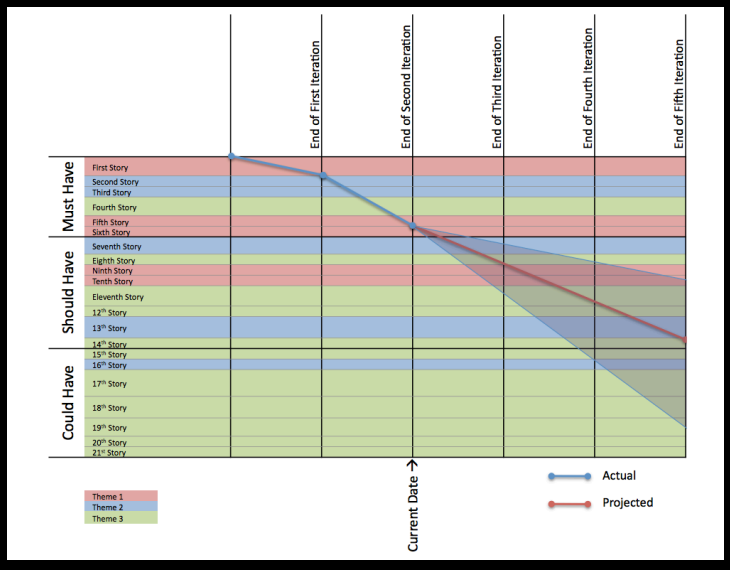

DON'T LET YOUR
AGILE MEASUREMENT OPPORTUNITY
BURN AWAY

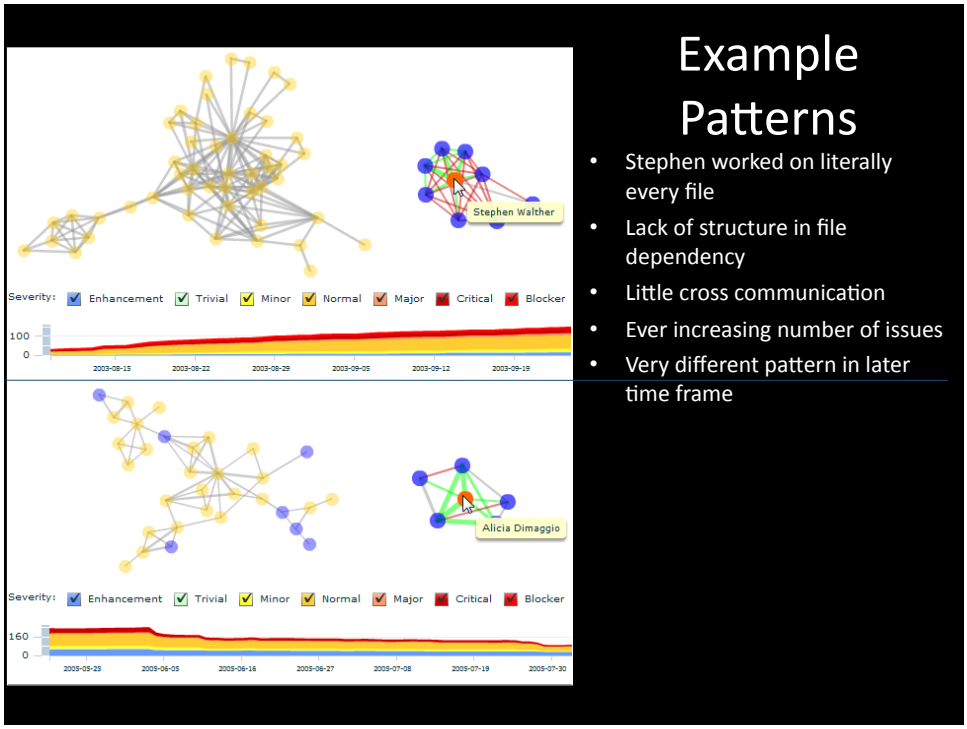

?

VISUALIZATION



Example cone of uncertainty





Example Patterns

- Stephen worked on literally every file
- Lack of structure in file dependency
- Little cross communication
- Ever increasing number of issues
- Very different pattern in later time frame

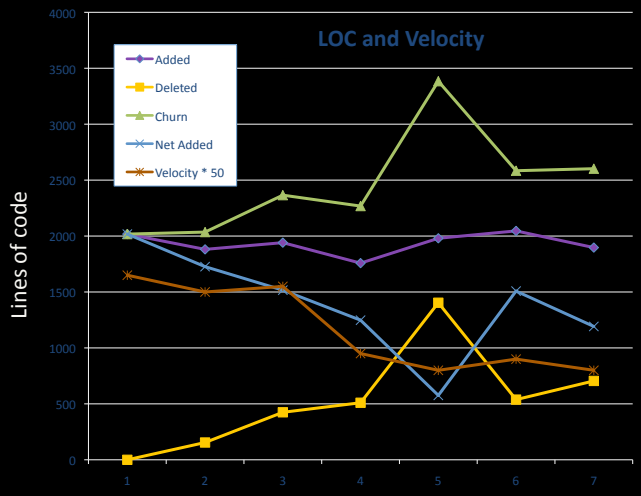
Example team decisions

- Should we do inspection? Should we do test driven development (TDD)? How much of each and in what combinations should we do?
- Should we live with a code base that has accumulated significant technical debt or should we refactor it? Or re-write it from scratch?
- Is geographical distribution going to cause high levels of defects? What can be done to mitigate this?
- What changes in architecture and team structure lead to minimal effort wasted on coordination?
- What should we do next?
- To what completion date should we commit?

Example exploratory data analysis

- Scenario
  - Team at Intuit recently started Scrum
  - Just finished 7th sprint (1 sprint = 1 month)
  - Velocity has slowed considerably
  - Will miss commitment unless velocity picks up
  - Code has accumulated technical debt
- Can we afford to refactor?
- Start by looking at data passively gathered in source code repository

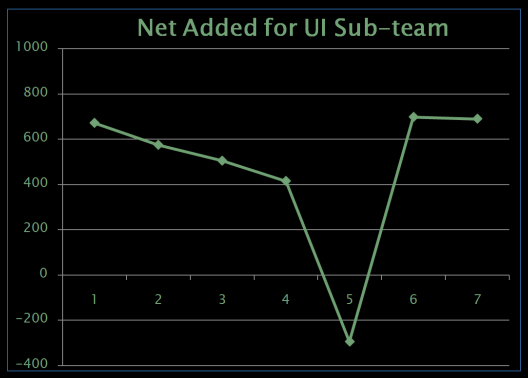
Project data



- Anomaly in 5th iteration
- That's when the UI sub-team did a refactoring
- Velocity correlates with Net Added LOC (R^2 = 0.75)

UI sub-team

- Head first design patterns book reading group
- Refactored to apply what they learned
- Code more "enjoyable" but was the refactoring worth it?
- Will it be worth it to refactor the rest of the code in iteration 8?



What to take away from example
(and not take away)

- Had to make a decision
- Was probably better to make it with some data than with none
- Used tacit knowledge
- Found their own pattern/predictive relationship
- If we can make it easy for teams to do this on their own and iterate rapidly, they are likely to find useful measures
- What not to take away:
  - General correlation between net added and velocity
  - That my research is focused on finding general predictive relationships (although it may very do that)