

Accounting for Agile

Agile project management for the bean counters

Brett Ansley
@pbansley
brett@brettansley.co.uk

The Problem

- Technology creates software
- Finance needs to recognise the software as an asset
- Costs and revenues expected from an asset need to be budgeted
- The value of assets needs to be recorded accurately

What Finance wants

- Budget
- Recognise
- Reconcile
- Amortise

The Rules

Or some of them anyway

- Any increase in the value of an asset (capital expenditure) must be auditable
- After an asset is “released” and starts generating value then additional work must be recorded as maintenance (operating expenditure) and must also be auditable
- The amount of time that a developer works on building an asset is directly related to allowable capital or operating expenditure attributable to the asset on the balance sheet

Valuing software

- People do work
- People work for a time
- Therefore the amount of time a person works = attributable value

Outcome of this thinking

- Timesheets for all developers
- Individuals who work the longest hours deliver the most value
- Resource optimisation around cost and hours

Reality

- Teams deliver value
- Teams that work together are more valuable than individuals
- Measuring individuals not teams reduces team effectiveness

Examples

- Timesheets are not accurate (and everyone knows it!)
- Project stuffing
- Project burying
- Management by inaccurate information

Behaviours

- Developers defer inputting timesheets
- Meetings / cross project assistance / coaching etc. are not captured in official statistics
- Timesheets always have the “right” number of hours
- Timesheets are demotivating for technology

Lessons

- Timesheets are a really bad way of capturing information
- Self reporting reduces effectiveness and increases focus on the individual
- We should stop using individual time based recording of attributable value

Different approach

- We already capture all the information we need for financial reporting
- We measure the output of a team (we use throughput or velocity)
- We have (relatively) stable teams so we know how much they cost

Recognise

- Reported % throughput (velocity) for each team broken down by feature
- Actual throughput (or velocity) are not reported outwards
- Reported who worked on the team during the period
- Work on a feature is reported as a percentage of total effort

Week 1	Team A	Team B	Team C
Feature A	20%		4%
Feature B	80%	16%	
Feature C		84%	20%
Feature D			76%

CapEx / OpEx

- Capital expenditure vs. Operational expenditure
- All projects were flagged as Opex or Capex in project tracking software
- Bugs in production code were assigned either Opex or Capex based on how long the feature had been in the market (initial bugs can be capitalised as part of the initial launch of the feature)

Budget 1st pass

- If only one team is working on the feature then:
- Team sizes project (by assigning points / breaking into stories / work units)
- The average throughput (velocity) of the team over the previous 5 periods is used and expressed as a range plus minus 2 standard deviations

Estimated cost_{low} = total size / average throughput_{high} * average team cost

Estimated cost_{high} = total size / average throughput_{low} * average team cost

- Because average throughput has 2 values (high and low) so estimated cost has 2 values (high and low) and is expressed as a range
- It is these values that are provided to stakeholders for costing

Budget 1st pass

- If multiple teams are working on the feature then:
- The work is broken down between the two teams and they each perform the above work
- The total estimate is the sum of the output of the two teams

Outcome of this refinement

- Better conversations about the value of a new feature
- Less insistence that teams finish exactly on budget
- Better understanding of variability by stakeholders

Reconcile 1st pass

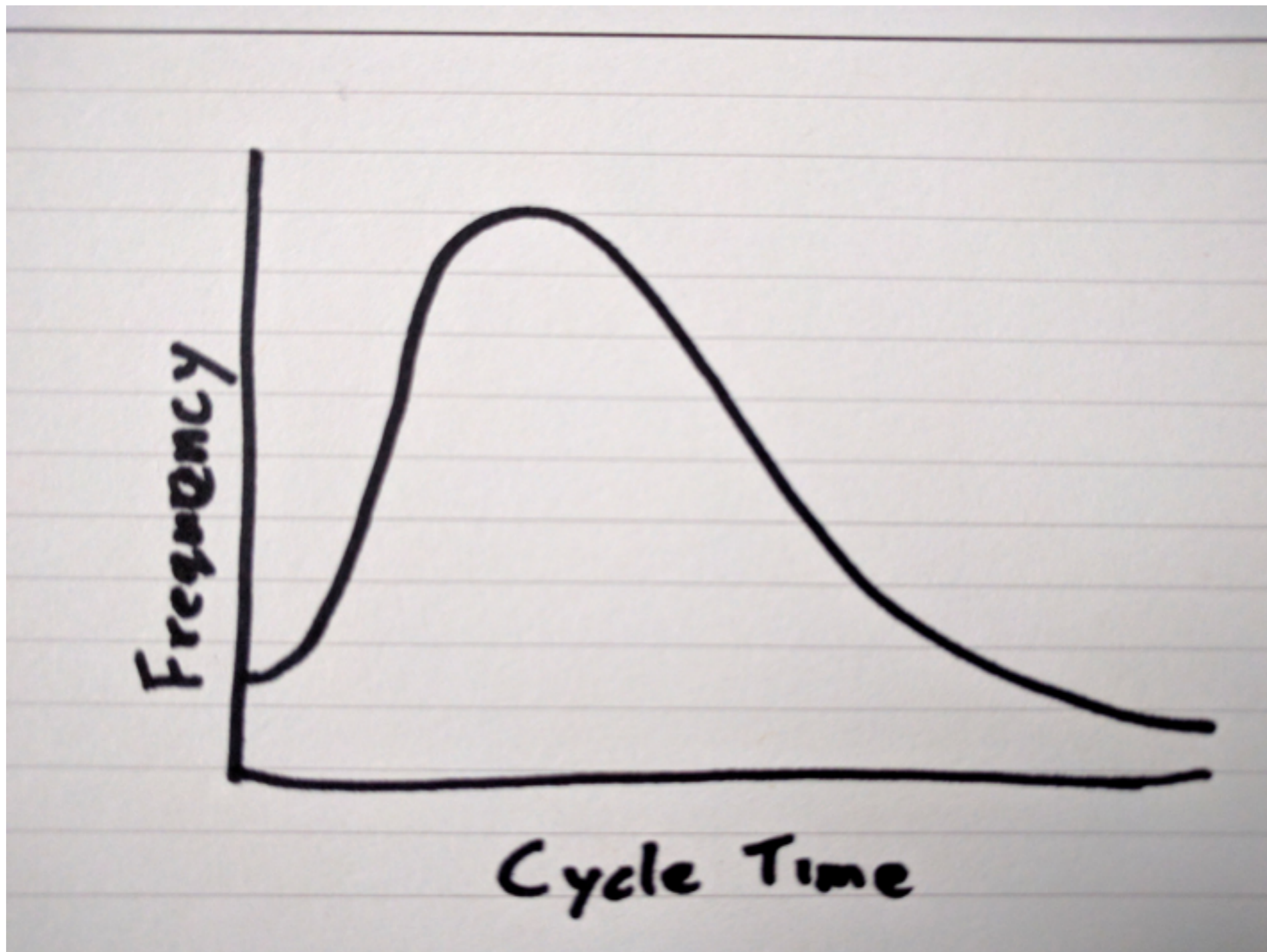
- We used the same formula for this calculation.

Remaining cost_{low} = remaining size / average throughput_{high} * average team cost

Remaining cost_{high} = remaining size / average throughput_{low} * average team cost

- Again expressed as a range

Noticed something



Budget 2nd pass

- If only one team is working on the feature then:
- Team sizes project (by breaking into stories / work units or likening it to other similar projects / features)
- Run a Monte Carlo simulation using a stochastic model of the development process

Estimated cost_{low} = 65 percentile duration * average team cost

Estimated cost_{high} = 85 percentile duration * average team cost

Reconcile 2nd pass

- We used the same process for this calculation.

Remaining cost_{low} = 65 percentile duration * average team cost

Remaining cost_{high} = 85 percentile duration * average team cost

- Again expressed as a range

Amortise

- Attribution across clients
- Life-Time value of feature
 - Uptake
 - Revenue
 - Longevity

Amortise

- Probability modelling of different scenarios
- Monte Carlo simulation to determine statistically probable model

What Tech reported

- At the end of each period each team reports:
 - Features worked on & bugs fixed
 - Percentage of effort expended on features & bugs (No. completed stories / actual throughput)
 - Team members during the period and number of days they worked
 - Outstanding work (expressed as a range) for all features in current WIP
- Each time a new feature is estimated the team reports:
 - Estimated cost of feature (expressed as a range)

How Finance used it

- At the end of each period finance can calculate:
 - Cost of features
 - Projected cost of features in progress
 - Projected cost of future feature development
 - Delta between original budget and current estimate

Satisfying the Auditors

- At the end of each period auditors can:
 - review project tracking software
 - review staff assignments
 - review project breakdown

Benefits

- Allows reporting at any point during feature development
- Allows reporting of features when they overlap during a reporting period
- No timesheets only minimal manual reporting from Technology
- Fully auditable
- Satisfied the original requirements from Finance

Brett Ansley

Agile coach
CPO C-Change

@pbansley
brett@brettansley.co.uk

