

Project Management

Project Management Agenda

- What it is
- What it should be

A Project

- Work out what to build
- Build it
- Bask in GLORY.

Project Management

- How to manage?
- How to report progress?
- (How to 'measure' progress)?
- How to ensure coordination?
- How to prioritise and adapt to changing priorities?
- How to maximise progress?

Estimation

- How long to get to the top?

?



Building software is like building bridges.



Please call me

- Next time you need a £20,000,000 spreadsheet!
- But even so...



The Millennium bridge



The Wobbly bridge

- Cost £18.2M
- Was 2.2M over budget (10%)
- And 2 months late
- Opened for 2 days
- Closed for 2 years
- Remediation cost £5M (30%)

Operator Error



2 worlds

- Process and management focused
- V
- User and delivery focused

- Process obsessed folks can become tool driven ('I need estimates in hours, the planning tool can't use days or weeks)'

Process management

- Too much time and effort wasted seeking the impossible (how long 'it' will take)
- Hint – if you are estimating in hours for something that takes more than a week you are dreaming.
- Minimum unit should be days, or weeks
- 1 day = 0.5 days to 1.499999 days (4hrs – 11.9999hrs)
- 8 hours = 7.5 hours to 8.4999
- 1 Calendar day = 0.6 developer days (on average)
- 1 developer day =approx= 2 calendar days
- Know which one your estimates are in

Agile

- Often means develop now, think later
- But really means work with what you know, know there is lots you don't know.
- Think, but don't over think, plan, but don't over plan.

Known Knowns

- On a multi dev month project the known knowns are irrelevant.
- The unknown unknowns will have a bigger impact than the other 3 put together

Spreadsheet Projects

- My ones anyway
- Between a few hours and 3 months, generally
- Usually single dev so coordination generally less critical
- Usually deliver 'something' on time
- Scope always expands during dev. This should be expected, welcomed and managed.

A 3 month Project (1)

- Several days
 - scope it out with user_, get a feel for the real business benefits
 - Broad calendar estimate (+/- 100%)
- Several weeks
 - Design overview – word doc with picture, max 4 pages (incl list of main components and interactions)
 - Maybe some super simple prototypes of complex elements
 - Dev days estimate (+/- 50%) (with time style caveat)
 - Agree priorities for various elements

A 3 month Project (2)

- A month or so
 - Build a little test a little
 - Deliver each component as soon as it is not (too) embarrassing
 - Before each delivery write the user guide
 - Use that exercise to enhance the component
- A couple of weeks
 - Fixing issues from early live use
 - Create developer documentation for the maintainers
 - Mainly by cleaning up code, refactoring for simplicity
 - Module summaries, sheet summaries
 - Dev tool set if required (wbStructure)
 - Gardening style general tidy up
 - Search code for TODOs.
 - Sensible names, and error handling should be there from start.

A 3 month Project (3)

- For ever
 - Address or deflect scope creep issues
- Well at least until the budget runs out..

On Being late

- Its a nightmare
 - Things that were put off to try and hit the deadline now really really need doing
 - Holidays that were delayed now need taking
 - The team is too run down to continue sprinting
 - Business needs will change
 - There will be a reorganisation
- Don't ever be late, deliver something.

Questions ?