Agile Project Planning and Management

Carl Erickson
Atomic Object LLC
Plan is a Noun

Success requires a plan
  – heavyweight upfront document
  – created at point of maximum ignorance
  – worked backwards from fixed dates often pulled from thin air
  – often created with little technical staff input
  – estimates made without requirements
  – close cousin of traditional, sequential processes
Natural Result

Plan is treated as holy writ
- significant investment in time
- linear and logical
- shows a path to successful completion
- tracking progress according to the plan
- imposing discipline on developers
- change is not viewed positively
Plan is a Verb

“Plans are worthless, planning essential.”
– Dwight Eisenhower (American general and President)

When reality diverges from the plan
– hide reality, or adjust the plan?

Planning in agile dev
– so important, we do it continuously
Planning

The hardest problems are…

– not technical
– involve people
– at least proportional to team size
Legitimate customer fears

Won't get what they ask for
Will ask for the wrong things
Pay too much for too little
Won't see a meaningful plan
Won't know where the project really stands
Won't be able to react to changes in the business
Legitimate developer fears

Will be told to do more than they know how to
Will be told to do silly things
Will have hard problems to solve without help
Will be given responsibility, but no authority
Won't be given clear descriptions of what is needed
Will have to sacrifice quality to a deadline
Won't have enough time to succeed
Results of Fear

Unproductive meetings
Memos and notes ("you said on Dec 12, ...")
Unwillingness to share information
Coverups
Building walls
Withdrawal of talent
Disengagement emotionally
No One Knows Everything

Business people know
  – scope
  – priorities
  – dates

Technical people know
  – how long things take
  – consequences
  – process
Planning Addresses Fear, Merges Knowledge

It is not anti-change spray
– stuff happens

Embracing change
– knowing reality vs tracking a plan
– always work on most important thing
– coordinate people
– understand the consequences of change
Why We Plan

To address legitimate fears

We plan in order to
  – work on the most important thing
  – coordinate people
  – understand the consequences of change

We don’t plan in order to
  – control events
Failures in Planning

Planning by activity, not feature
Lateness accumulates
Activities are not independent
Applications are not developed by priorities
Confusing estimates with commitments
Activities vs Features

Planning by activity, not feature
– customers don’t care about activities
– it’s hard to spot missing features when the plans are in terms of activities
– activities can’t be judged “done”
– activities don’t finish early (Parkinson’s Law)
Lateness Accumulates

Activities tend to finish late, but never early

Traditional planning focuses on the dependencies of activities
- Gantt charts

Downstream activities are at risk
- they probably can’t start early
- they may be late themselves
- they inherit every previous schedule slip
Dependence not Considered

Example
– planning for 10 main web pages, 5 hours each for styling, 50 hours total
– what if the first one takes 10 hours?

Do you
– update the plan?
– assume you’ll get faster?
– assume you’ll get lucky?
Using Customer Priorities

Traditional plans are made in terms of activities or infrastructure
– customers don’t understand these
– customers don’t get value from these
– customers can’t evaluate these

When things get tight
– what gets dropped may be more important than what got built
Dealing with Uncertainty

The future is impossible to know
Software projects are risky and complex
  – or they shouldn’t be done
Implications of uncertainty
  – dates are really ranges
  – estimates include a probability
Don’t plan to get it all right, up front
  – iterate
Summary

Agile planning differs from traditional planning

– more focused on planning, not a plan
– embraces change
– is continuous