



~~#BeyondProjects~~

~~#NoProjects~~

#ProjectLess

Beyond Projects

*Or The End of Projects
and what happens next*

Allan Kelly - allan@allankelly.net

<http://www.allankelly.net>

[Twitter: @allankellynet](https://twitter.com/allankellynet)

TopConf

Linz

March 2017

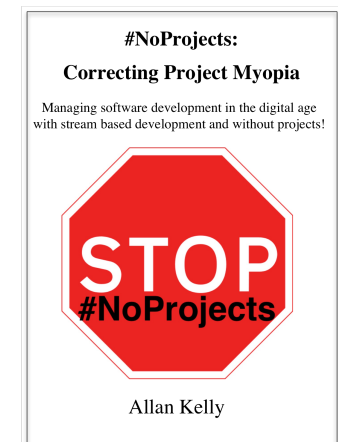
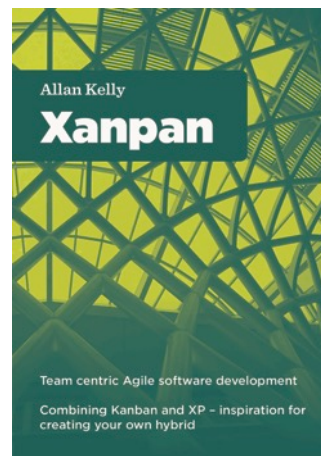


Allan Kelly...

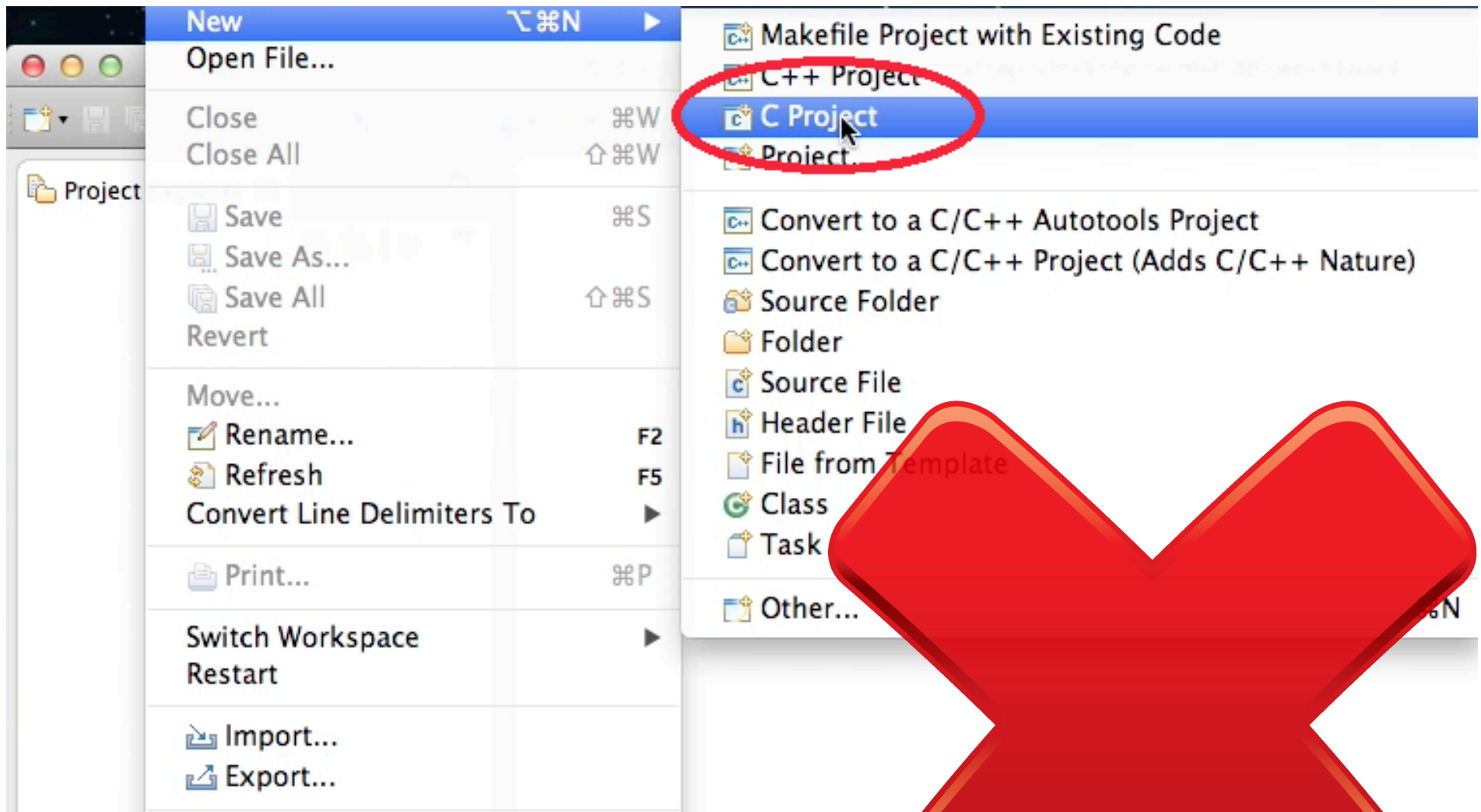
- Consulting on software development & strategy
- Training for Agile

Author

- **Little Book of Requirements & User Stories** (2016)
<http://www.leanpub.com/userstories>
- **Xanpan: Team Centric Agile Software Development** (2014)
<https://leanpub.com/xanpan>
- **Business Patterns for Software Developers** (2012)
- **Changing Software Development: Learning to be Agile** (2008)



What is a “project”?



Taken from <http://armstrap-documentation.readthedocs.org/en/latest/getting-started-eclipse-development-tools.html>

project

noun |'prɒdʒekt|

1 an individual or collaborative enterprise that is carefully planned to achieve a particular aim: *a research project* | *a project to build a new power station.*

- a piece of research work undertaken by a school or college student: *a history project.*

- a proposed or planned undertaking: *the novel undermines its own stated project of telling a story.*

2 (also **housing project**) N. Amer. a government-subsidized housing development with relatively low rents: *her family still lives in the projects.*

A project is....



"PMI defines a project by its two key characteristics:

- it is temporary and
- undertaken to create a product, service, or result that is unique."

Project Management Institute - <http://pm4id.org/1/2/>

Problem #1 – Success?

Project Success Criteria

- On Schedule
- On Budget
- On Quality (Features)

Where's the
value?
£ \$ €



Project Model Assumptions

1. Value is knowable

And is known before start

2. There is no value in flexibility

i.e. Options are valueless



Requirements change.

The observed rate of change
in the US is about 2% per
calendar month

Compound to
~27% per annum

Capers Jones, 2008

The world changes

Get over it

IT/IS & Business Manager survey

70% believe they are failing to identify and quantify the benefits adequately

38% openly admit they overstate the benefits in order to obtain funding

80% report that the review and evaluation of completed projects is also inadequate due to the focus on [achieving] **cost, time** and **quality** objectives and not on whether the intended **benefits** were **realized**.

Delivering value from IS and IT investments, John Ward, Cranfield School of Management, 2006 – survey of UK & Benelux managers

<http://www.som.cranfield.ac.uk/som/dinamic-content/research/documents/deliveringvaluereport.pdf>

The Project
model leads
to...



Conflict and....

Goal displacement

- Chasing date over benefit
- Chasing time over benefit
- Chasing cost over benefit
- Chasing features over benefit

Solution



Governance based on
actual delivered
benefits

Iterative portfolio
management

Focus on Benefit not The End

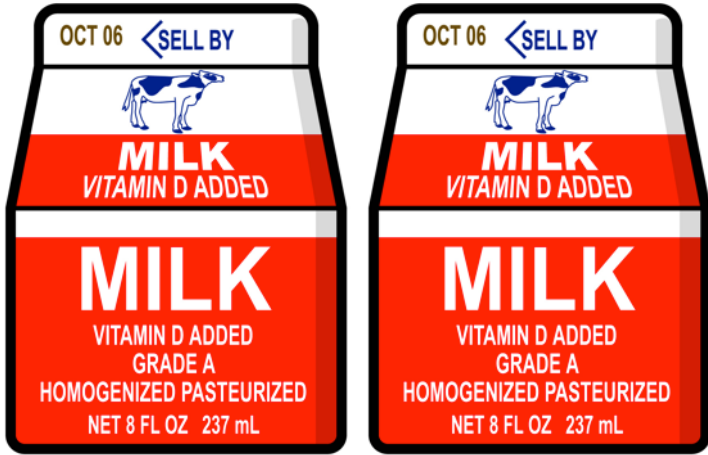
Ask not, “When will the software be done?”

Do ask: “When will the software deliver value next?”

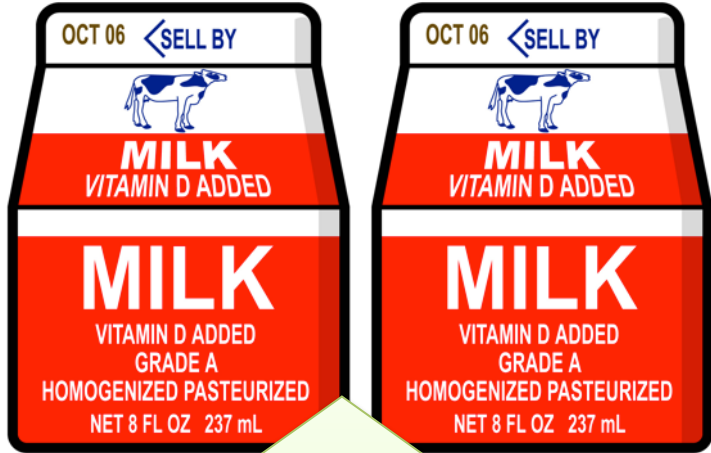


Problem #2 – Projects are Big

- Project model is optimized for big
- Used on small pieces of work it inefficient
- Projects push big decisions up
 - to big men
 - with big cheque books
 - top-down authority



Milk is cheapest in BIG cartons



And small cartons of software reduce risk

Software is cheapest in lots of small cartons

Big increases
risk

Project A: Risk = 30% Value at risk = £1m
Therefore risk weighted value = £300,000

Consider a large project

Prj B: Risk = 15%
Value @ risk = £½m
Therefore ... = £75,000

Prj C: Risk = 15%
Value @ risk = £½m
Therefore ... = £75,000

E: Risk = 6%
@risk = £200k
Therefore = £12k

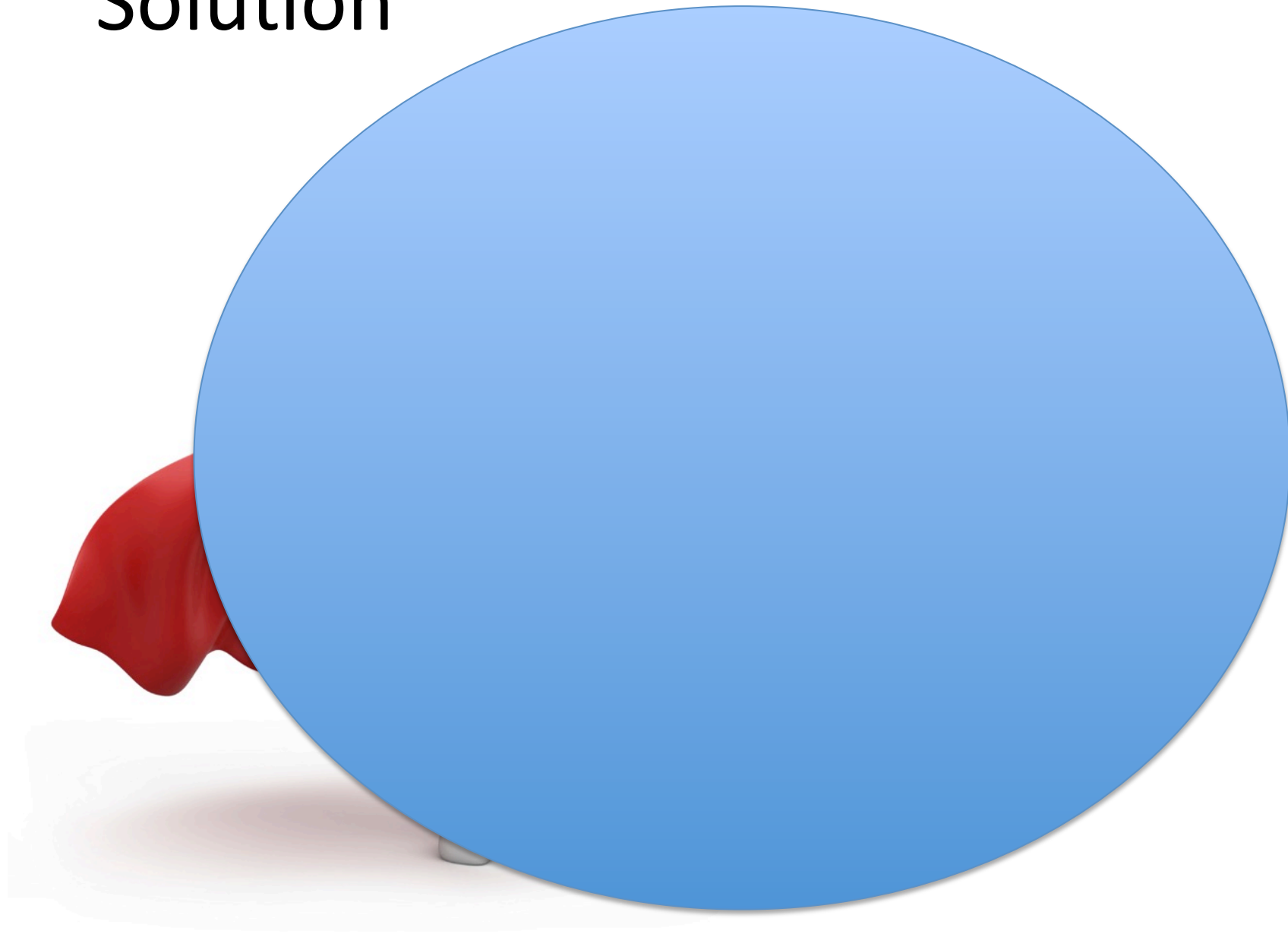
F: Risk = 6%
@risk = £200k
Therefore = £12k

G: Risk = 6%
@risk = £200k
Therefore = £12k

H: Risk = 6%
@risk = £200k
Therefore = £12k

I: Risk = 6%
@risk = £200k
Therefore = £12k

Solution



Software development...

- Does NOT have economies of Scale
- Development has DISECONOMIES of scale

Therefore

- Stop thinking **BIG**
- Start thinking SMALL



Projects are big batch

Small batches best for software

Small batches reduce risk

Regular delivery increases ROI

Problem #3 – The End

Software isn't temporary

Successful software doesn't stop

Successful software continues to change

Only dead software has an end-date

Projects end
Successful software
doesn't



Successful software?

Search Results for "Web Browser" Sort By: Relevance

Showing page 1 of 13.

Moodle ENTERPRISE
Moodle is a Course Management System (CMS), also known as a Learning Management System (LMS).
23,239 weekly downloads

Moodle is a Course Management System (CMS), also known as a Learning Management System (LMS). It is a Learning Environment (VLE). It is
<https://moodle.org/>

Last Updated: 3 days ago

4.7
★★★★★

Download Now

Web Torrent
Webtorrent is a web based GUI written in JavaScript for Bit Torrent Library. It is designed for server applic...

Webtorrent is a web based GUI written in JavaScript. Installation of desktop bt clients are not required.

Last Updated: 2013-04-17

Weekly downloads: 0
Last update: 17 April 2013 (9mths)

PerlLORD
Web wrapper for the up...
0 weekly downloads

Web wrapper for the up...
program converts ANSI to telnet client or anyt

Last Updated: 2013-02-25

Weekly downloads: 0
Last update: 25 May 2013 (8mths)

1) If they use it, it will change

2) Only Dead Software Stops changing

Data from SourceForge search for "WebBrowser" 19 Jan 2014

End dates damage quality

- Short term thinking leads to...
 - Corner cutting
 - Known & unfixed bugs
 - Residual technical debt
 - Knowledge lost



A Project is...

“A **temporary organization** that is needed to produce a unique and **predefined outcome** or result at a **pre-specified time** using **predetermined resources.**”



PRINCE2 definition
of project

Continuous flow

Continuous improvement

Continuous delivery

Continuous benefit

Continuous not Temporary

Software
Development



Project
Management

A Match Made in Hell

Software is forever

Projects are
TEMPORARY



Problem #4 -

Temporary Organizations

Temporary Team?

- Forming
- Storming
- Norming
- Performing
- Destroying



Takes time & money!



*Why destroy performing teams?
Why spend that money?
Why loose knowledge?*

Temporary organizations

The most destructive idea known to software development

Disbanding teams destroys

- Knowledge
- Capability
- Performance

Solution

Base work around
stable **stream teams**

Aim for continuity

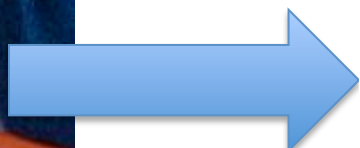
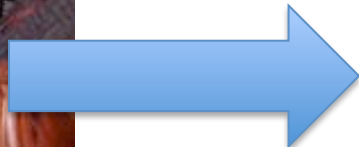
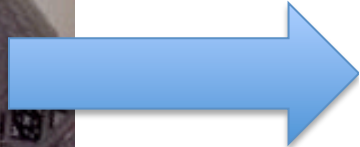


Bring the work to the team

Bring the work to the team

Bring the work to the team

Bring the work to the team



What to do about it...

- Keep teams together
- Flow work to the teams
- Work in the small
- Work continually
- Demonstrate value



The End of Projects



- Projects are accounting codes
- Finished Software is Dead Software
 - Living software changes
 - Dead software doesn't
 - Living software doesn't end
- Project thinking kills software

Beyond Projects

A new model....



Waterfall 2.0 Continuous Flow



Jonathon's Run Fall, Pennsylvania by Hubert Stoffels (<http://flickr.com/photos/22195940@N00>)
Creative Commons License

Get good at small

Optimize & Organize for lots of small

Deliver small increments of value

And evaluate results



Stream Teams

Continuity in teams

Align with business

Product

Service

Business Line

Devolve authority:

Efficiency in making lots of
small decisions



Value seeking & delivering

Value seeking

Identify

Deliver

And EVALUATE

Governance of teams

Repeat, don't stop



Could this ever work?

- Governance by results?
- Fail fast, fail cheap?



Feedback

Option A



HOME CONFERENCE ABOUT
SPEAKERS REGISTRATION PARTNER
SCHEDULE CONTACT

TOPCONF
Linz

TRACK: **AGILE**

Beyond Projects #NoProjects – why projects are wrong and what to do instead

Good projects make for bad software. The defining feature of a project is an end date, the defining feature of successful software is that it doesn't end. Software which is useful is used and demands change, stop changing it and you kill it. At best the concept of a "Project" is erroneously used for software development work. At worst the Project metaphor leads to...

Option B

1/2 price code:
Linz2017



~~#NoProjects:~~ **#ProjectLess**

Correcting Project Myopia

Managing software development in the digital age
with stream based development and without projects

Allan Kelly



#NoProjects / #ProjectLess

- It ain't ever over

allan kelly - allan@allankelly.net

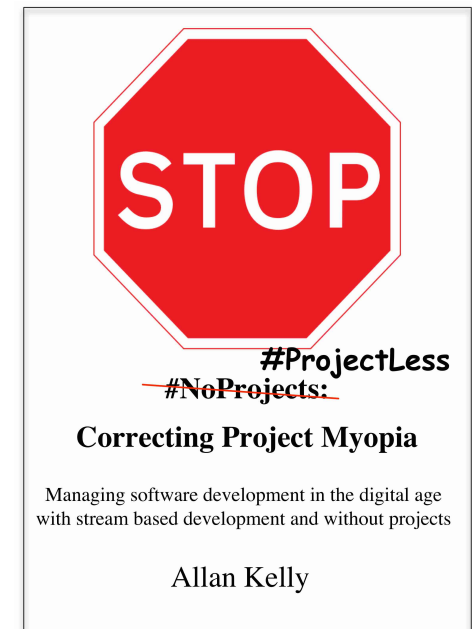
Twitter: [@allankellynet](#)

1/2 price code:
Linz2017

#NoProjects

LeanPub

<https://leanpub.com/noprojects>



Appendix

End Date considered harmful

Late requirements considered
inferior



Change Governance

Base Governance on **actual**
delivered benefits

Not milestones completed

Not documents

Not budgets

What have
you delivered
for me lately?

Picture from Picasa - Creative Commons License
[http://commons.wikimedia.org/wiki/
File:House_of_Parliament_6_2012-07-08.jpg](http://commons.wikimedia.org/wiki/File:House_of_Parliament_6_2012-07-08.jpg)



Governance based on results

- Experiment friendly
 - Failure tolerant
 - Fail fast, fail cheap (salvage)
- Align with strategy
- Governance through *Iterative Portfolio Management*



Iterative Portfolio Management

- Fixed schedule for gate reviews
- Review what is ready on a date
 - Benefit delivered to date
 - Potential benefit if continued



Rational end dates?

In the US more than half of the large projects ... predetermined end date is selected, and it is forced on the project by arbitrary decree.

Capers Jones, 2008



Solution

Work to the business
deadline

Do what you can in the
time

Do good work



Corporate Psychopathy

Process by which corporations
disband performing teams and
release staff

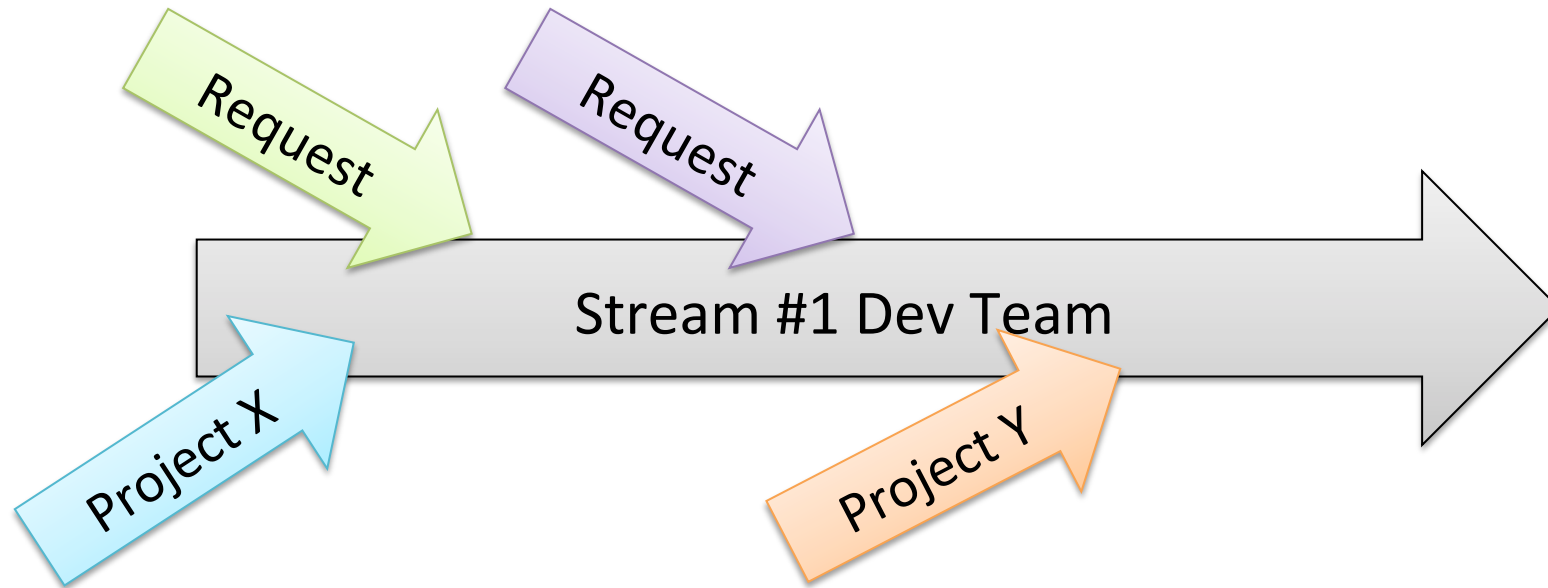


Solution

Organize work around
that which is stable



Organize by business stream & team



- Aim for stable teams & continuity
- Close to business
- Manage queues within capacity

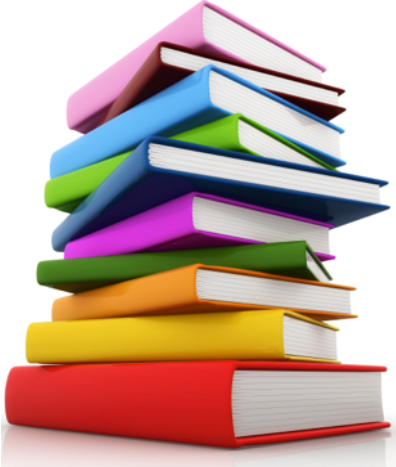
Deadlines Good



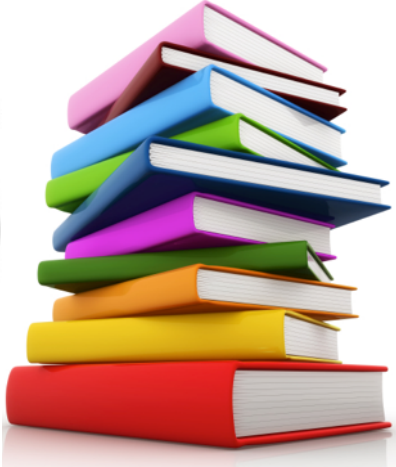
End dates Bad



Big Batch Size



Wait



Wait

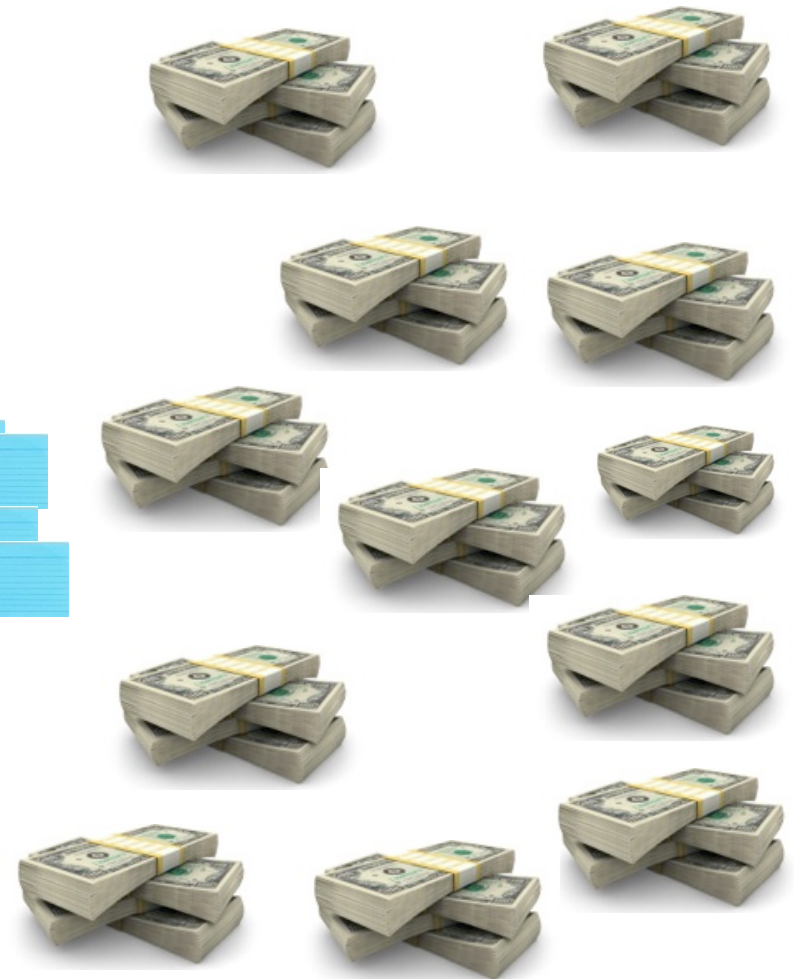
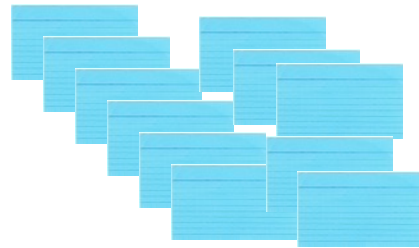
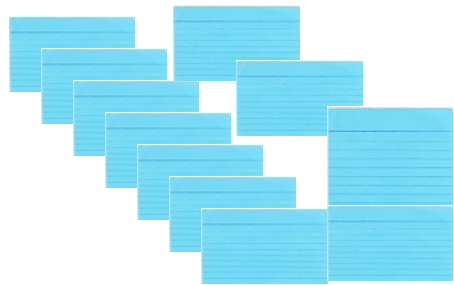


Build this!

Deliver this!

Make lots of this!

Small Batch Size



... and keep going

Not fair!
You misrepresent us!



Project Manager



Wait a minute....



So what is “A project”?

Why does Prince 2 contain this definition?

And what does a “Project Manager” actually *Manage*?



We don't
Break up teams, or
Stick doggedly to
requirements

We do

- Allow change
- Consider value
- Work continually

Project Manager

Problem #5: Language

“False Projects”

When the language of projects is used to discuss work which is not a project



Confusion!

**Stop confusing people
Stop misleading
yourselves**