The Use of Cadence in Agile and Traditional Projects

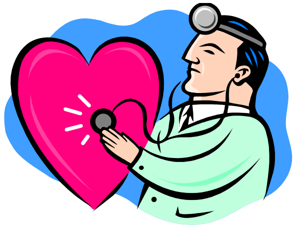
Sprint Length, Integration Frequency, and Other Matters of Rhythm

Kathy Iberle

March 14, 2013

Rose City SPIN

Agile Today



- “The team works for a fixed period of time called a sprint.” Ken Schwaber; 2002
- “A regular iteration rhythm acts like a heartbeat for the project.” Mike Cohn; 2006

Fixed-length iterations are
an example of *cadence*

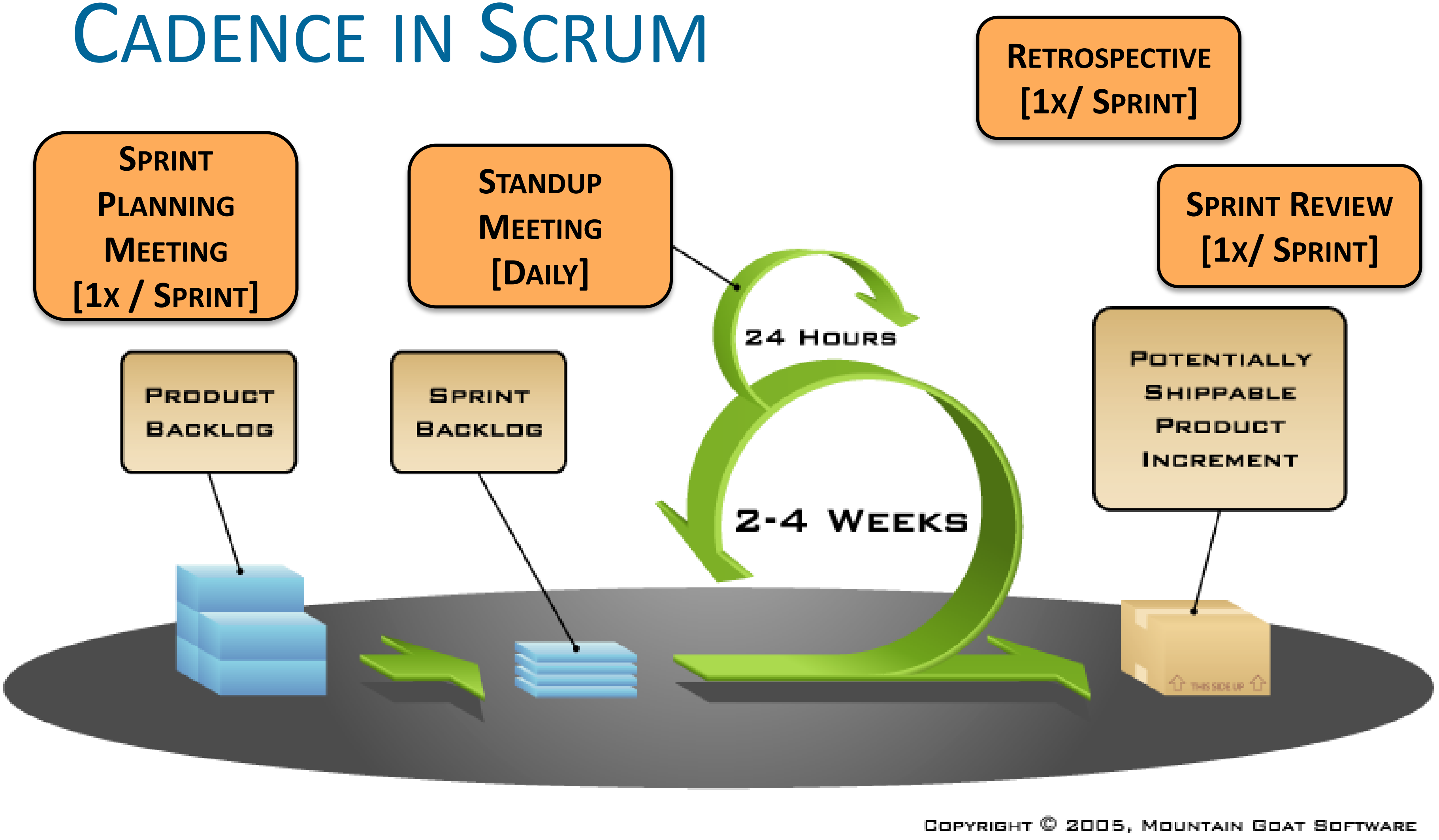
Cadence

- *Cadence*: a regular, predictable rhythm within a process.

- Cadence increases efficiency by cutting out duplicated or avoidable work.

IBERLE
CONSULTING GROUP, INC.

Cadence in Scrum



Retrospective [1x/ Sprint]

Sprint Planning Meeting [1x / Sprint]

Standup Meeting [Daily]

Sprint Review [1x/ Sprint]

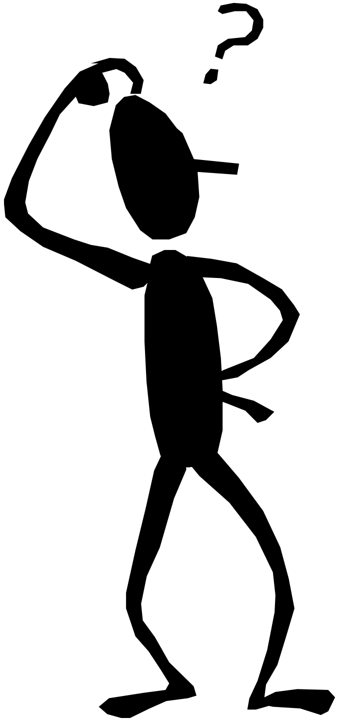
Copyright © 2005, Mountain Goat Software

Graphic courtesy of
Mountain Goat Software

IBERLE
CONSULTING GROUP, INC.

WHAT IS A GOOD CADENCE?

- Not too frequent, not too infrequent
- That's nice – can we be any more exact?

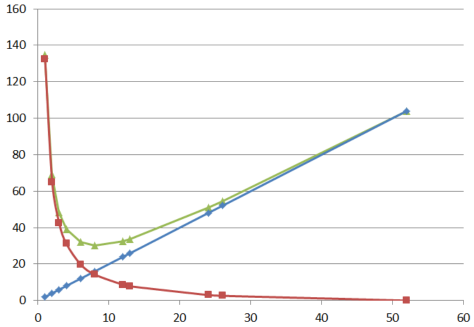


IBERLE
CONSULTING GROUP, INC.

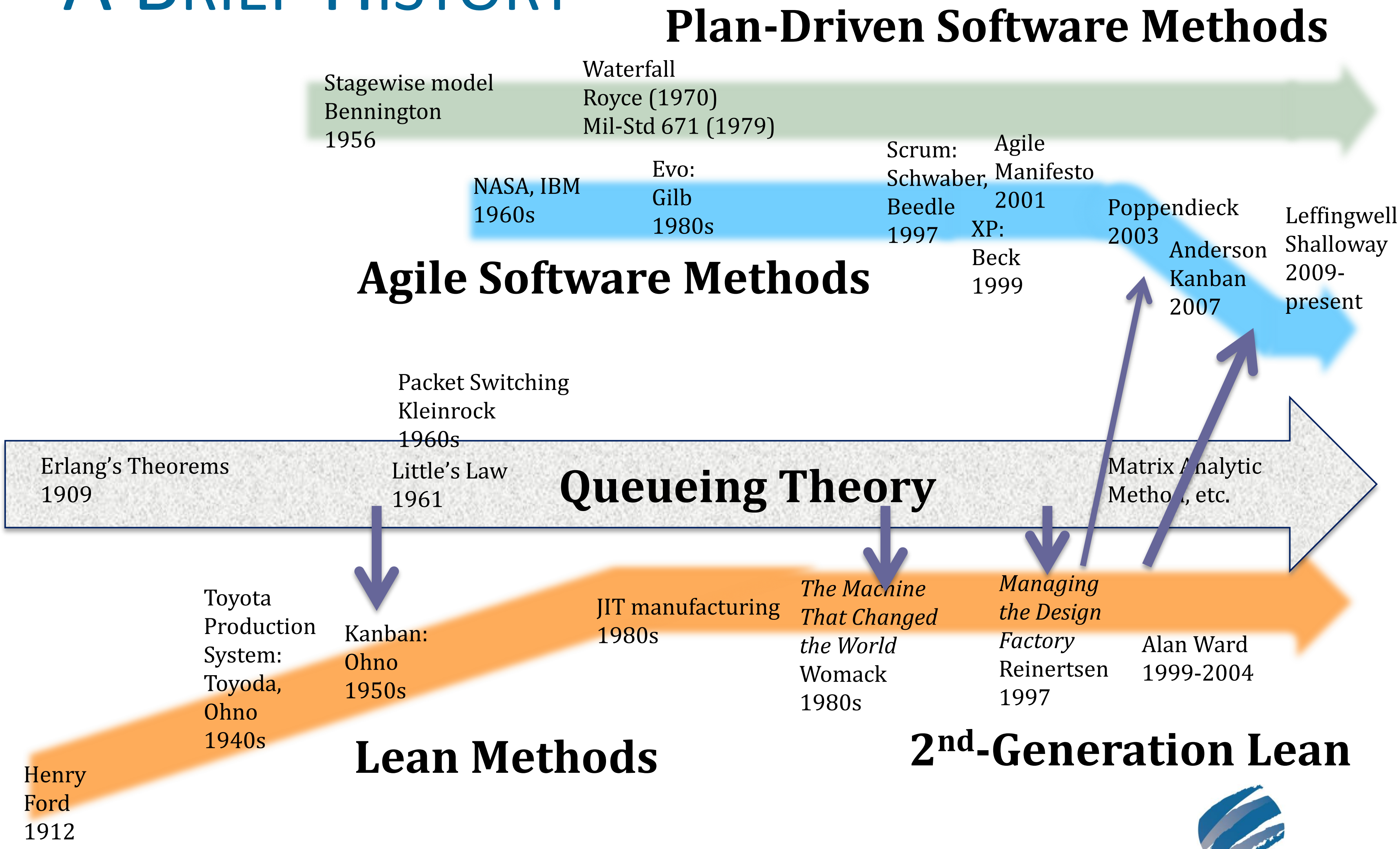
A Scientific Way to Look at Cadence

1. Identify the benefits of the cadence
2. Identify the cost of the cadence
3. Apply mathematics to find optimal point

Fortunately, once you've seen the math, you rarely need to do actual calculations



A Brief History



Plan-Driven Software Methods

Stagewise model
Bennington
1956

Waterfall
Royce (1970)
Mil-Std 671 (1979)

Agile Software Methods

NASA, IBM
1960s

Evo:
Gilb
1980s

Scrum:
Schwaber,
Beedle
1997

Agile
Manifesto
2001

XP:
Beck
1999

Poppendieck
2003

Anderson
Kanban
2007

Leffingwell
Shalloway
2009-
present

Queueing Theory

Erlang's Theorems
1909

Packet Switching
Kleinrock
1960s

Little's Law
1961

Matrix Analytic
Methods, etc.

Lean Methods

Henry
Ford
1912

Toyota
Production
System:
Toyoda,
Ohno
1940s

Kanban:
Ohno
1950s

JIT manufacturing
1980s

The Machine That Changed the World
Womack
1980s

Managing the Design Factory
Reinertsen
1997

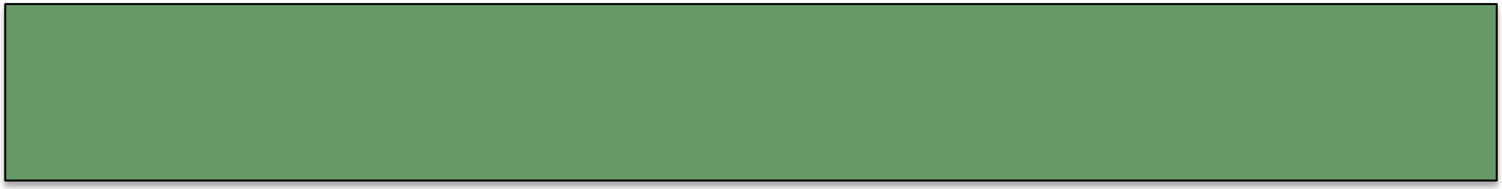
Alan Ward
1999-2004

2nd-Generation Lean

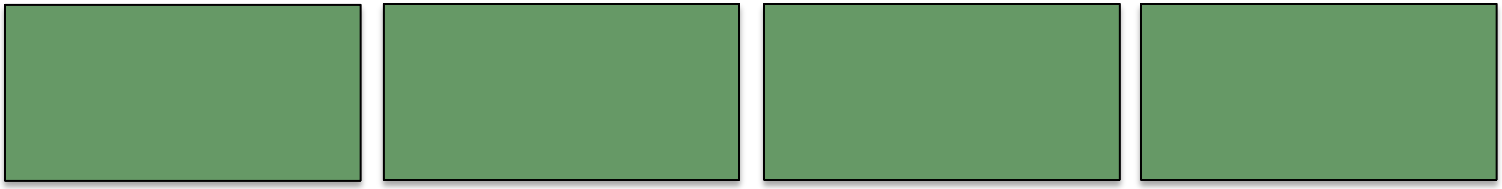
IBERLE
CONSULTING GROUP, INC.

SPRINTS SPLIT THE WORK INTO BATCHES

- Waterfall: no stopping points



- Agile: stopping point after each sprint



What do we *lose*
if there are no stopping points?

The Cost of Longer Sprints

Holding Cost includes the cost of:

- Rework due to not getting earlier feedback from customer on whether features are satisfactory
- Extra effort to fix bugs because they weren't found earlier
- Lost sales due to not having the chance to change or add a feature based on new information
- Lost profit because we didn't ship, sell, or otherwise use any features before the end of the project

More Sprints => Lower Holding Cost

IBERLE
CONSULTING GROUP, INC.

Holding Cost Accumulates

Cost of holding one month's work for one month = 🟧

If we continue for a year without stopping:

Holding cost for first month's work
is paid 11 times 🟧🟧🟧🟧🟧🟧🟧🟧🟧🟧🟧

IBERLE
CONSULTING GROUP, INC.

HOLDING COST ACCUMULATES

Cost of holding one month’s work for one month = ■

If we continue for a year without stopping:

Holding cost for first month’s work
is paid 11 times

■■■■■■■■■■■

Holding cost for second month’s work
is paid 10 times

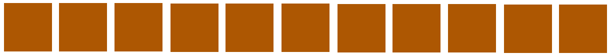
 ■■■■■■■■■■
■■■■■■■■■■■

Holding Cost Accumulates

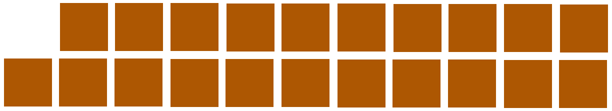
Cost of holding one month's work for one month = 🟧

If we continue for a year without stopping:

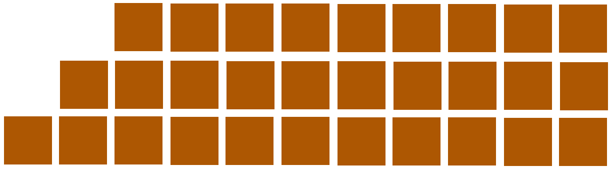
Holding cost for first month's work
is paid 11 times



Holding cost for second month's work
is paid 10 times



Holding cost for third month's work
is paid 9 times

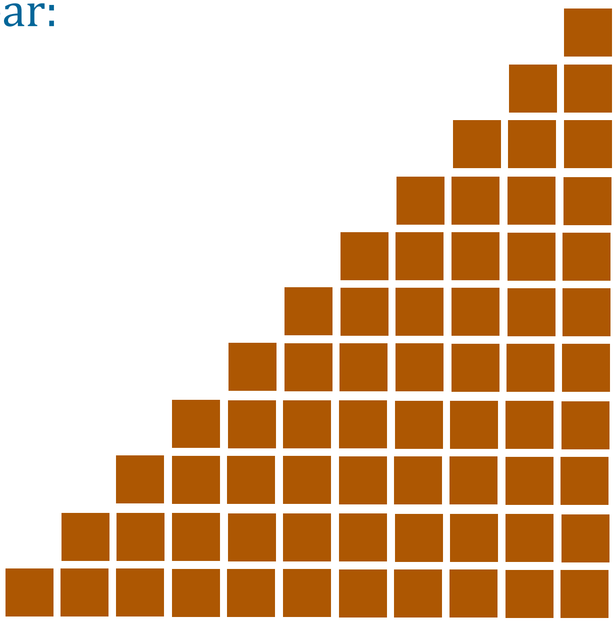


Holding Cost Accumulates

Total cost of holding the work for an entire year:

$$= H * \sum_{i=1}^{11} i \quad = \; H * 66$$

Where H = cost of holding one month's work for one month

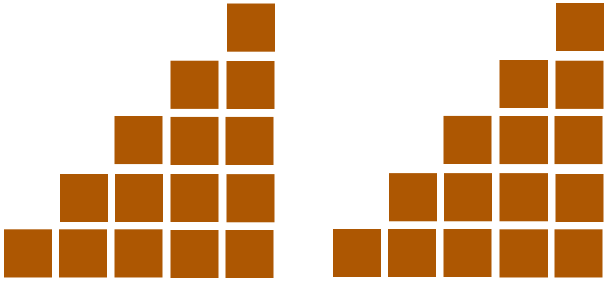


© 2013 Iberle Consulting Group, Inc.

The Benefit of Sprints

Total holding cost for six month sprints

$$= H * 2 * \sum_{i=1}^{5} i \; = \; H * 30$$

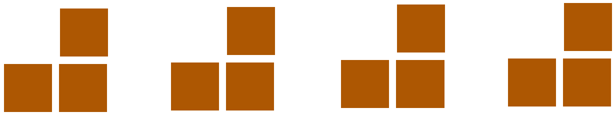


IBERLE CONSULTING GROUP, INC.

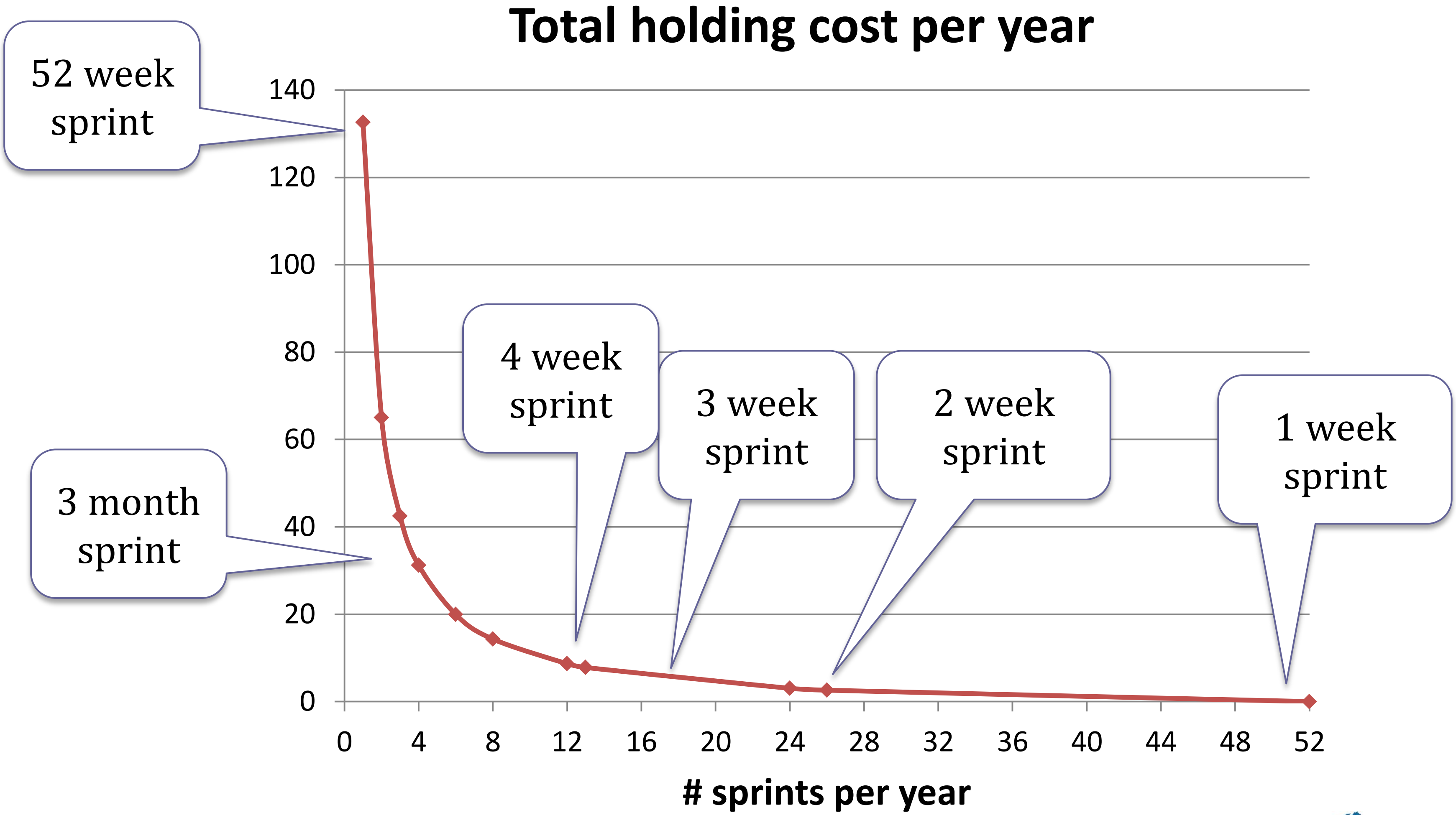
The Benefit of Sprints

Total holding cost for three month sprints

$$= H * 4 * \sum_{i=1}^{2} i \; = H * 12$$



Total Holding Cost



IBERLE
CONSULTING GROUP, INC.

The Cost of Short Sprints

What does it cost each time we start or end a sprint?

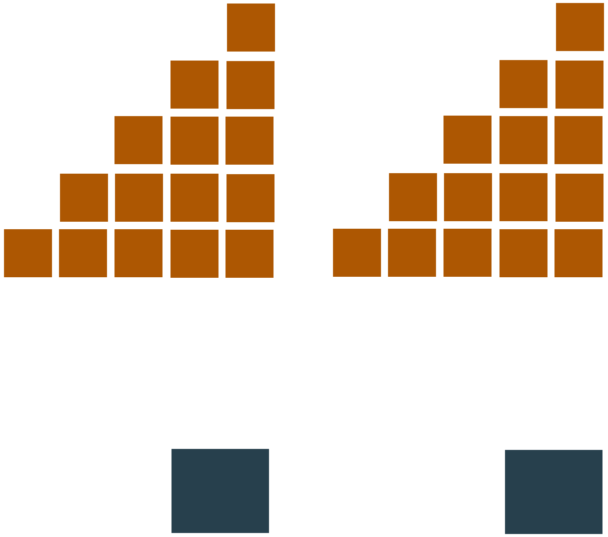
Transaction Cost – Short Sprints

What does it cost each time we start or end a sprint?

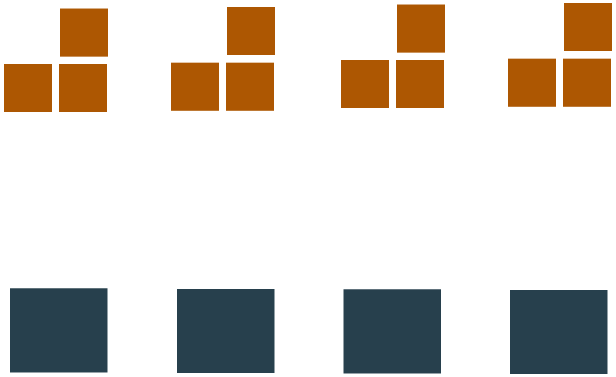
- Running system integration and test
- Holding sprint review and sprint planning meetings
- Final defect review & decision-making

- This is called the *transaction cost*

Transaction Cost for Shorter Sprints

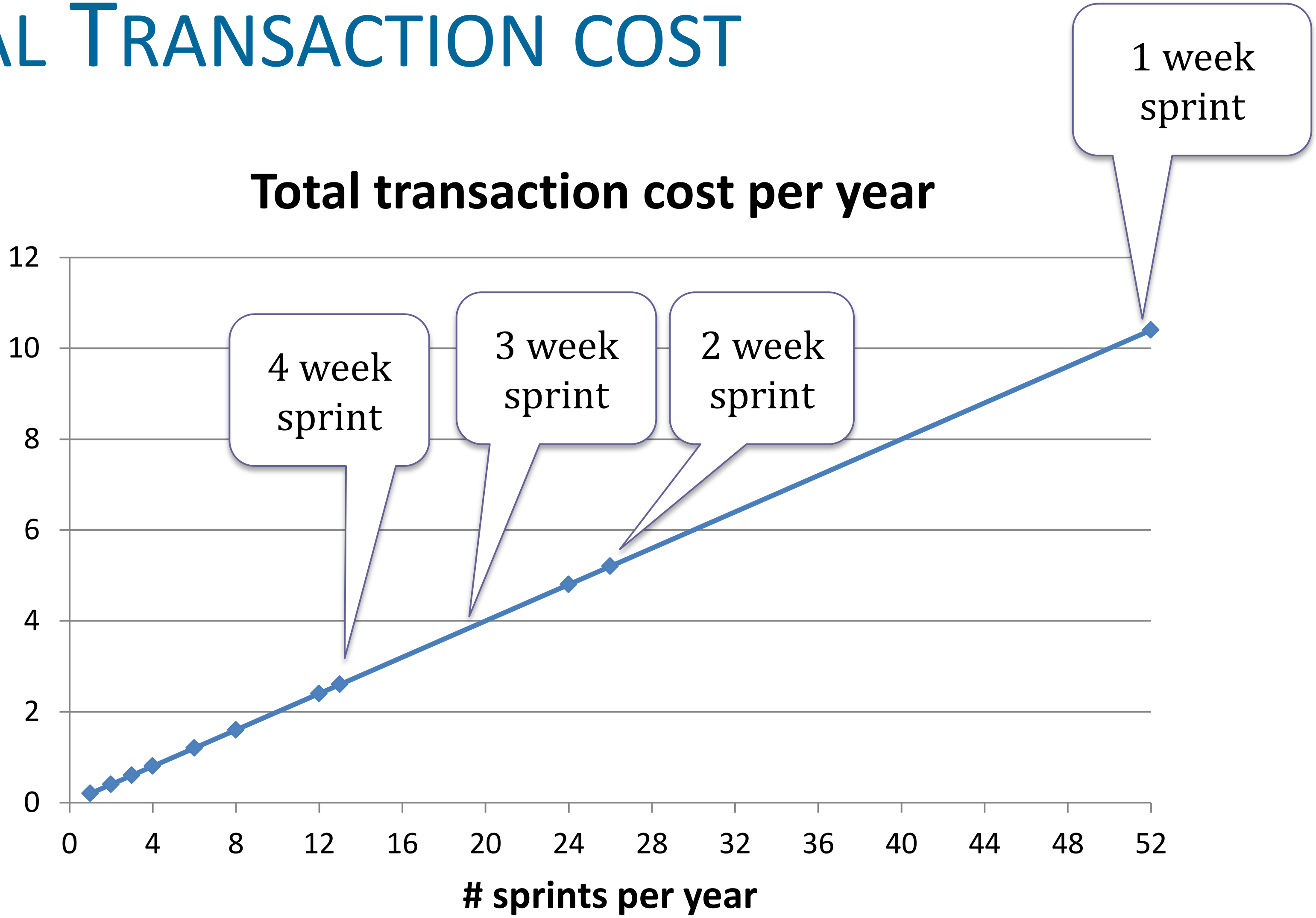


Total transaction cost
for six month sprints:



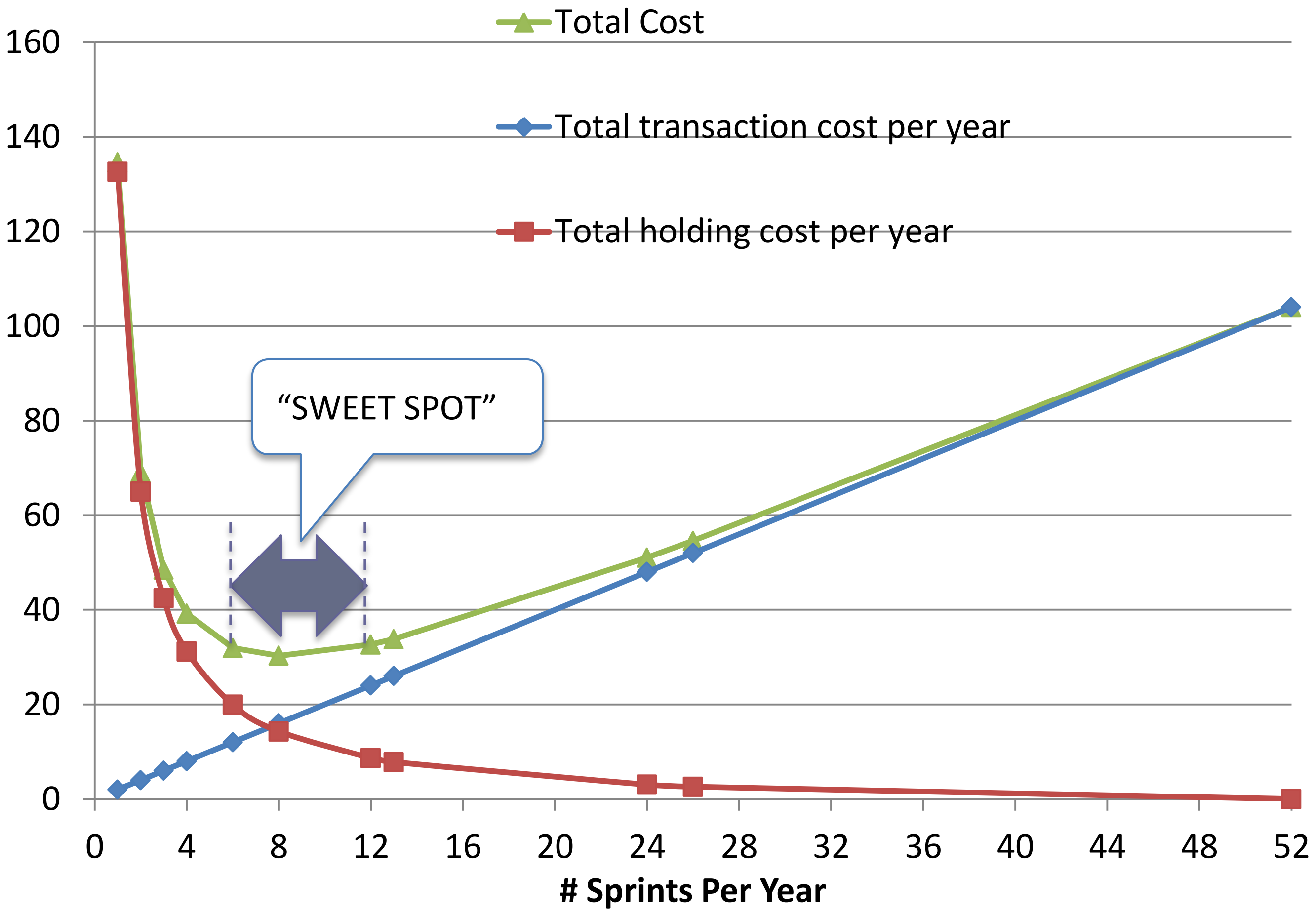
Total transaction cost
for three month sprints:

Total Transaction cost



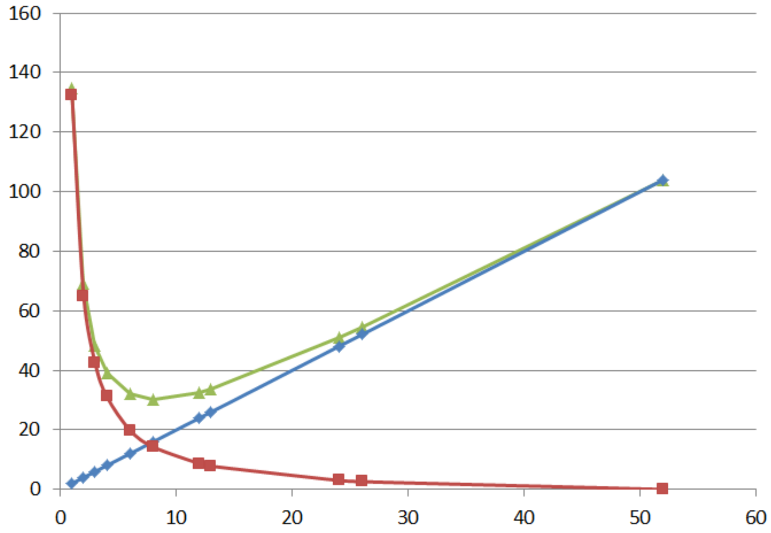
IBERLE
CONSULTING GROUP, INC.

Total Cost = Holding Cost plus Transaction Cost



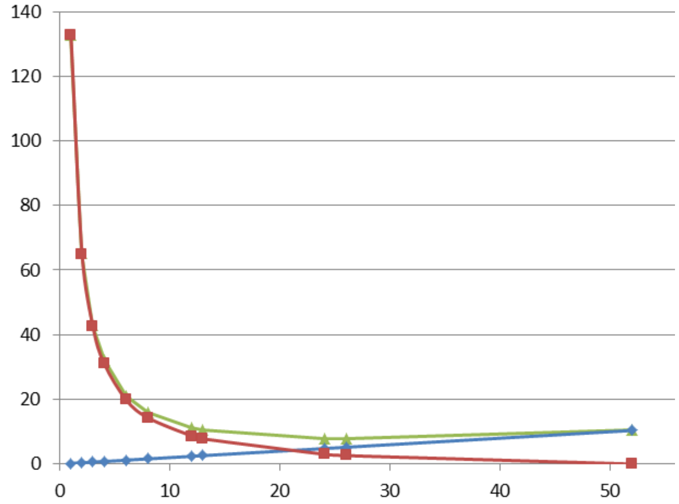
IBERLE
CONSULTING GROUP, INC.

CHANGING THE TRANSACTION COST



T = 2

Sweet spot = 6 week sprints



Reduce transaction cost by 90%

T = 0.2

Sweet spot = 2 week sprints

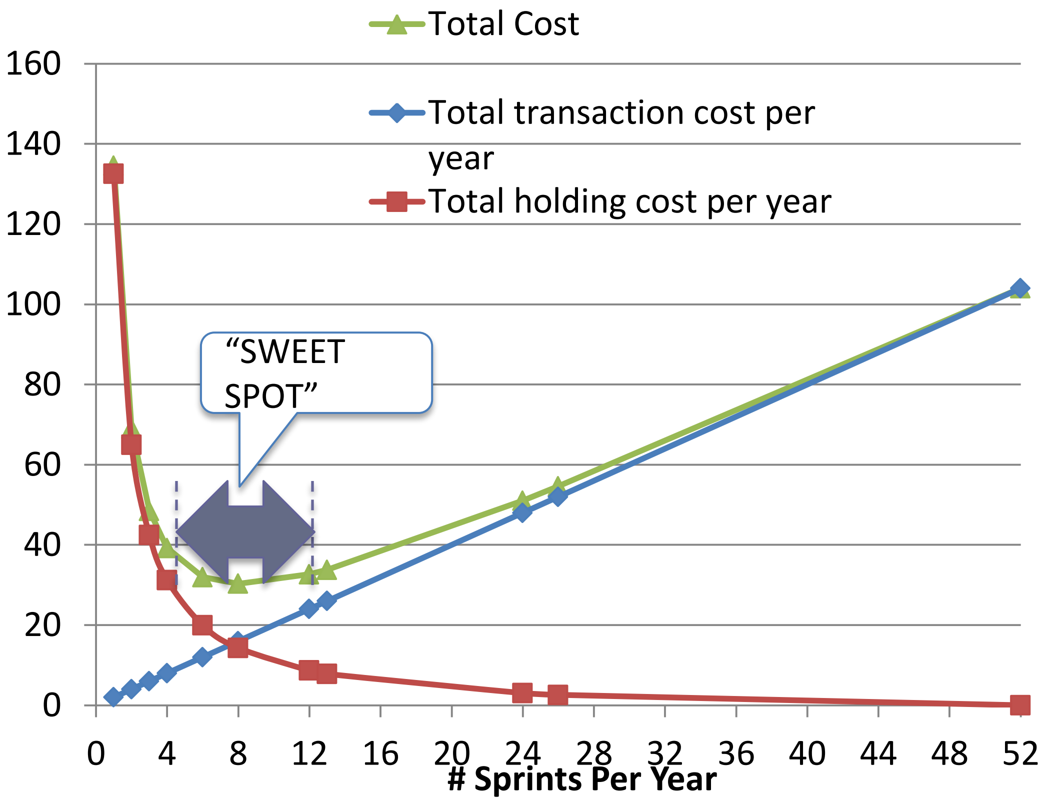
IBERLE
CONSULTING GROUP, INC.

WHAT THIS MEANS TO US

- Big gains when moving from 1 sprint/year to 6 sprints per year

- After that, exact sprint length doesn't make nearly as much difference ….

- Unless the transaction cost is dramatically reduced



"SWEET SPOT"

© 2013 Iberle Consulting Group, Inc.

Mike Cohn on Sprint Length

- How long priorities can remain unchanged
- Willingness to go without outside feedback
- The amount of uncertainty

Holding Cost

- Overhead of iterating
- Ease of getting feedback

Transaction Cost

From Mike Cohn's book *Agile Estimating and Planning*

IBERLE
CONSULTING GROUP, INC.

2ND-Generation Lean View

- Rework due to not getting earlier feedback from customer on whether features are satisfactory
- Extra effort to fix bugs because they weren't found earlier
- Lost sales due to not having the chance to change or add a feature

Holding Cost

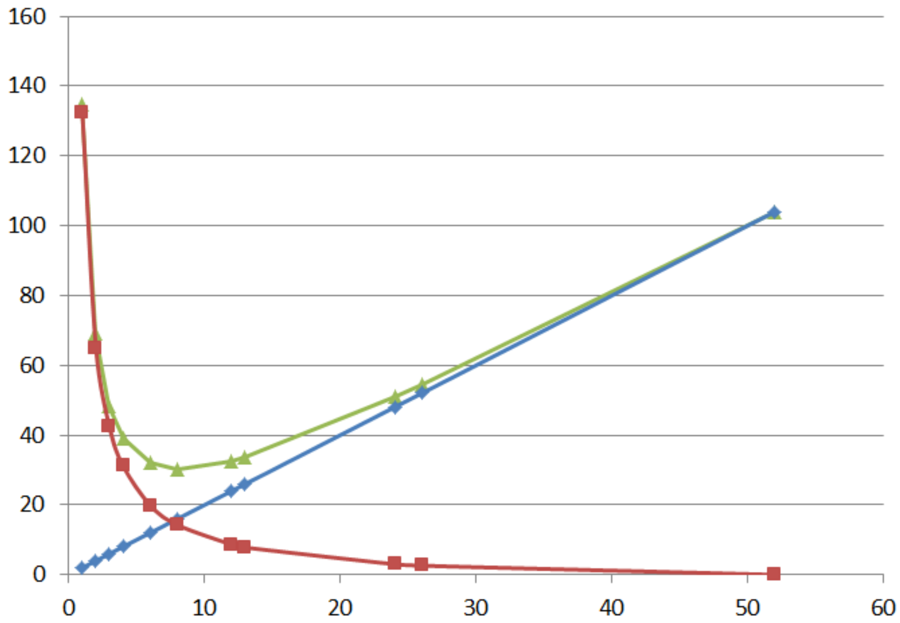
- Cost of system integration and test
- Cost of holding sprint review meetings
- Final defect review & decision-making

Transaction Cost

From Mike Cohn's book *Agile Estimating and Planning*

Continuous Integration

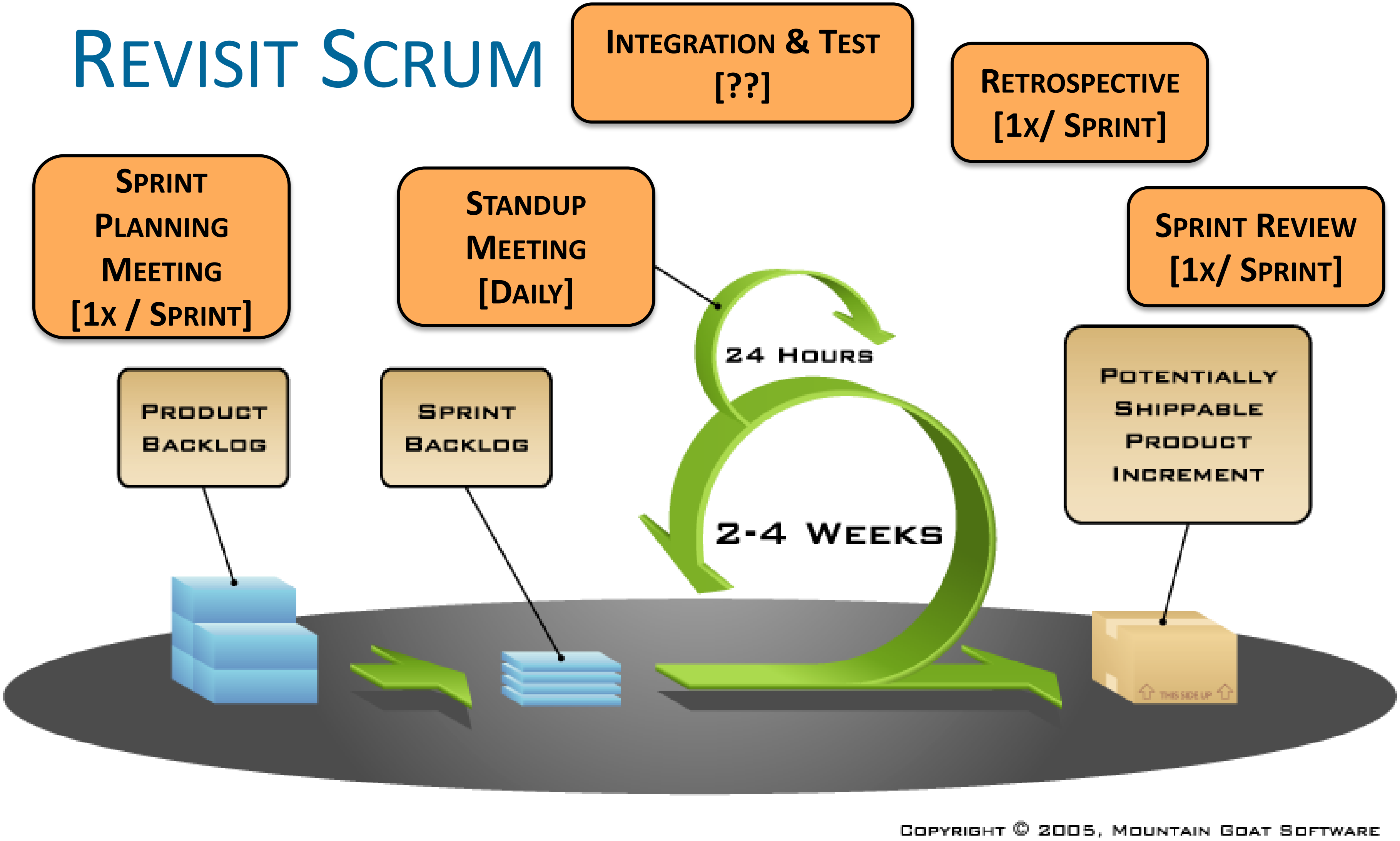
- Not immune to the cost trade-off!
- What are the holding costs of integration?
- What are the transaction costs?



CONTINUOUS INTEGRATION

- There is a sweet spot for integration
- Consider these transaction costs:
  - Check-in, including required pre-check-in tests
  - Running the integration tests
  - Diagnosing defects or errors reported by tests – including false positives!

Revisit Scrum



Integration & Test [??]

Retrospective [1x/ Sprint]

Sprint Planning Meeting [1x / Sprint]

Standup Meeting [Daily]

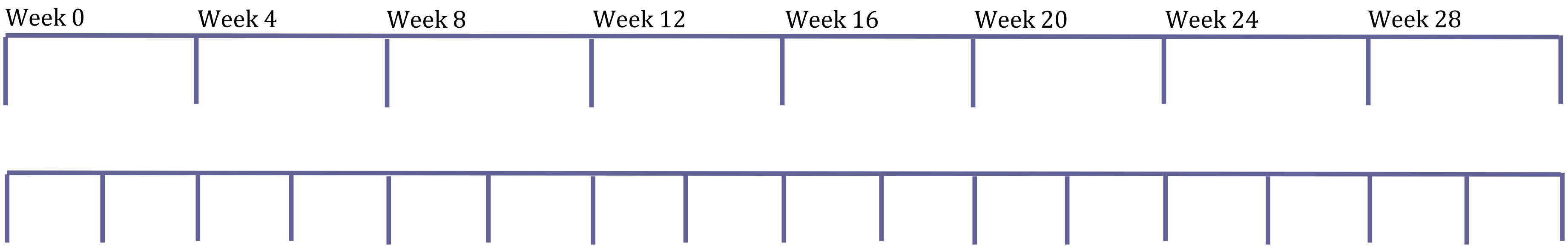
Sprint Review [1x/ Sprint]

Copyright © 2005, Mountain Goat Software

Graphic courtesy of
Mountain Goat Software

ALIGN YOUR CADENCES

- Cadences for the same project should be multiples of each other
  - For instance, retrospective every other sprint



THE IMPORTANCE OF PREDICTABILITY

- The predictability of cadence is critical

Other Applications of Cadence

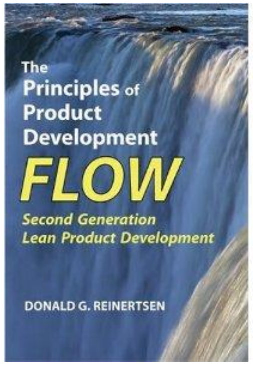
- Product introduction
- Use of Part-Time Resources
- Testing
- Status meetings
- Other ideas?

The Bottom Line

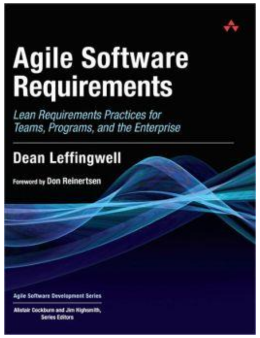
- The best cadence is the one which minimizes the total cost
- Total cost is driven by both holding cost and transaction cost
- The “sweet spot” for total cost is usually broad and flat
- Moving the “sweet spot” may require an order-of-magnitude reduction in transaction cost

REFERENCES ON CADENCE

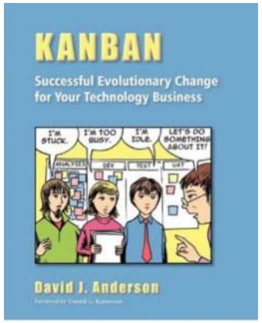
Everything you ever wanted to know about cadence.
The Principles of Product Development Flow: Second Generation Lean Product Development
by Don Reinertsen


The Principles of Product Development
FLOW
Second Generation Lean Product Development
DONALD G. REINERTSEN

Cadence to coordinate large projects
Agile Software Requirements : Lean Requirements Practices for Teams, Programs, and the Enterprise
by Dean Leffingwell


Agile Software Requirements
Lean Requirements Practices for Teams, Programs, and the Enterprise
Dean Leffingwell
Foreword by Don Reinertsen
Agile Software Development Series
Alistair Cockburn and Jim Highsmith, Series Editors

Decoupling the Scrum cadences
Kanban: Successful Evolutionary Change for Your Technology Business
by David J. Anderson


KANBAN
Successful Evolutionary Change for Your Technology Business
I'M STUCK.
I'M TOO BUSY.
I'M IDLE.
LET'S DO SOMETHING ABOUT IT!
ANALYSIS
DEV
TEST
UAT
David J. Anderson

Thank You!

This slide set is available at www.kiberle.com/articles

Learn more about cadence at www.kiberle.com/links

31ST ANNUAL
PACIFIC NW SOFTWARE QUALITY CONFERENCE
OCTOBER 14-16, 2013
THE MANY FACES OF QUALITY
learn • network • inspire

PNSQC 2013 Invited Speaker: *Lean in the Test Lab: The Potential for Big Improvements*
Workshop on Wednesday, October 16 – *Lean in the Test Lab – Work Smarter, Not Harder*

IBERLE
CONSULTING GROUP, INC.