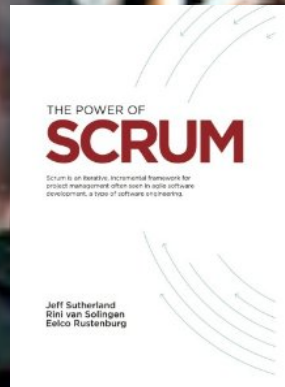


AGILE CONTRACTS: MONEY FOR NOTHING AND CHANGE FOR FREE

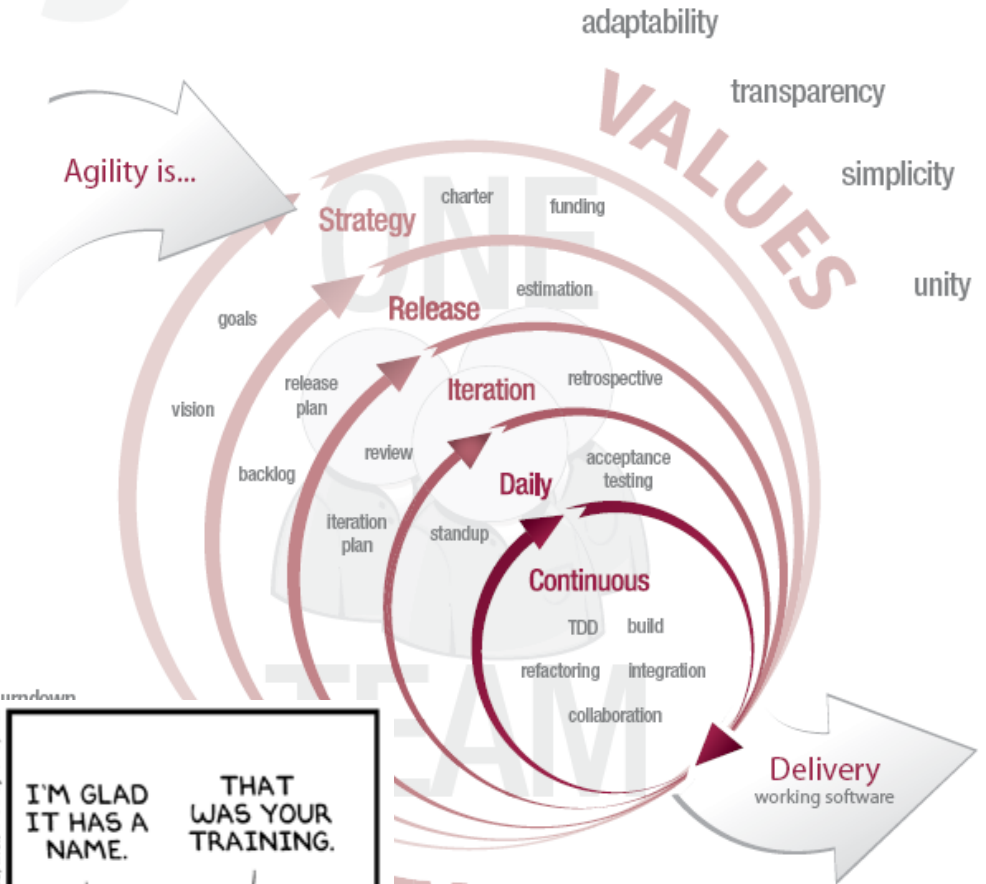


With help from Citrix Online, Google, Yahoo, Microsoft, IBM, Oracle, MySpace, Adobe, GE, Siemens, Disney Animation, BellSouth, Nortel, Alcatel-Lucent, EMC, GSI Commerce, Ulticom, Palm, St. Jude Medical, DigiChart, RosettaStone, Healthwise, Sony/Ericsson, Accenture, Trifork, Systematic Software Engineering, Exigen Services, SirsiDynix, Softhouse, Philips, Barclays Global Investors, Constant Contact, Wellogic, Inova Solutions, Medco, Saxo Bank, Xebia, Insight.com, SolutionsIQ, Crisp, Johns Hopkins Applied Physics Laboratory, Unitarian Universalist Association, Motley Fool, Planon, FinnTech, OpenView Venture Partners, Jyske Bank, BEC, Camp Scrum, DotWay AB, Ultimate Software, Scrum Training Institute, AtTask, Intronis, Version One, OpenView Labs, Central Desktop, Open-E, Zmags, eEye, Reality Digital, DST, Booz Allen Hamilton, Scrum Alliance, Fortis, DIPS, Program UtVikling, Sulake, TietoEnator, Gilb.com, WebGuide Partner, Emergn, NSB (Norwegian Railway), Danske Bank, Pegasystems, Wake Forest University, The Economist, iContact, Avaya, Kanban Marketing, accelare, Tam Tam, Telefonica/O2, iSense/Prowareness, AgileDigm, Highbridge Capital Management, Wells Fargo Bank, Deutsche Bank, Hansenet/Alice, GlobalConnect, U.S. Department of Defense, Agile Lean Training, EvolveBeyond, Good Agile, Océ, aragostTRIFORK, Harvard Business School, Schuberg Philis, ABN/AMRO Bank, Acme Packet, Prognosis, Markem-Imaje International, Sonos, Mevion, Autodesk, First Line Software, SCRUMevents, UPC Cablecom, NIKO, CWS-BOCO, BottomLine, Lean Enterprise Institute, Liberty Global, Samsung, Monster, Dartmouth University, Health Leads, Samsung R&D Center, JP Morgan Chase

Agile 2008 Toronto - updated at Agile 2013 Nashville

Agile Development

ScrumButt



Accelerate Success

Nokia Test

as updated by Citrix and Scrum Inc.

- As an agile citizen, I can assess a team's behavior and compare it to current Scrum best practices, so I can consider changes that might increase productivity.

v. 2013-7-31

Nokia-Citrix Test Origins

- Bas Vodde
 - 2005: Originates Nokia test to scale agile to all of Nokia Networks
 - 2007: Updates Nokia test for Scrum in Nokia/Siemens Networks
- Jeff Sutherland
 - 2007,8: Tunes Nokia test, establishes scoring system
- Dan Greening
 - 2011: Updates test at Citrix, recasts questions as stories

Instructions

- Each person on a team should have a piece of paper
 - There are 10 assessments
 - Each assessment has a score from 0 to 10
 - In each assessment, sum the "Acceptance Tests" scores that pass.
 - Total score will range from 0 to 100
- Average the scores for team members to get the team score

Assessment 1: Iteration

- As a team, before we commit to a Sprint, we know its duration, so we deliver better rhythmic, synchronized value.
- Acceptance Tests:
 - Variable, $4 < \text{duration} \leq 6$ weeks: 2
 - Variable, $\text{duration} \leq 4$ weeks: 4
 - Constant for last 3 iterations, $\text{duration} = 1$ month: 5
 - Constant for last 3 iterations, $\text{duration} = 4$ weeks: 6
 - Constant for last 3 iterations, $\text{duration} = 3$ weeks: 8
 - Constant for last 3 iterations, $\text{duration} \leq 2$ weeks: 10

Assessment 2: In-Sprint Testing

- As a team, we take joint responsibility for all testing, so our Sprint product has sufficient quality to be immediately deployable.
- Acceptance tests (sum):
 - Team creates some unit tests in-sprint: 1
 - Team creates unit tests for each story in-sprint: 1
 - Team tests each story prior to Sprint Review: 2
 - Team tests each story immediately after coding: 2
 - Team automates feature tests for each new story: 2
 - Build system packages, deploys to stage or live, and runs all automated feature tests at least every 24 hours: 2

Assessment 3: Sprint Stories

- As a team, we commit to work only when backlog items conform to a Definition of Ready, so we generate business value fast.
- Acceptance Tests (sum):
 - Sprint requirements are documented: 1
 - Requirements are independent, well-prioritized stories: 1
 - Stories start with this: "As a <stakeholder>, I can <do something>, so <business gains value>": 2
 - Stories have externally verifiable acceptance tests: 2
 - Team has a written, enforced Story Definition of Ready: 2
 - Team has a written, enforced Story Definition of Done: 2

Assessment 4: Product Owner

- As a team, a single Product Owner helps the team understand and prioritize value, so we generate profits long-term.
- Acceptance tests (sum):
 - A single external person (PO) prioritizes work: 2
 - PO interrupts team work only during Scrum meetings: 2
 - PO attends all Planning, Grooming, Review and most Standups: 2
 - PO creates a product backlog, with stories estimated by the team before Sprint Planning: 1
 - PO maintains a velocity-aware release roadmap: 1
 - PO motivates team to reduce technical debt: 2

Assessment 5: Product Backlog

- As a team, we have a value-ranked backlog, so we can focus on work that will generate the most business value for the least effort.
- Acceptance Tests (sum):
 - Team serves multiple prioritized Product Backlogs: 1
 - Team serves a single prioritized Product Backlog: 2
 - PO regularly discusses release burndown with team, and adjusts backlog priorities based on historic velocity: 1
 - Stories more than 3 months out trend larger in effort: 1
 - Team can explain the ROI of each story: 1
 - PO assesses value (NPV, buy-a-feature) to rank stories: 2
 - PO prioritizes cheap prototypes to test value early: 2

Assessment 6: Estimation

- As a team, our estimates are largely free of statistical bias, so stakeholders can rely on release forecasts and make more money.
- Acceptance Tests (sum):
 - Team agrees to estimates before committing: 1
 - PO, SM and non-developers do not supply estimates: 1
 - Team carefully avoids anchor bias before estimation: 1
 - Representatives or actual team creates poker estimates: 1
 - Actual team creates poker estimates: 2
 - Teams use reference stories to make estimates: 2
 - Actual velocity is $< \pm 20\%$ of estimated velocity: 2

Assessment 7: Sprint Burndown

- As a team, we know our progress toward completion of backlog items, so members can help with high-priority work-in-progress.
- Acceptance Tests (sum):
 - Burndown exists, team knows where it is: 1
 - Team reviews, adjusts tasks and burndown daily: 1
 - Tasks have hour or point estimates estimates (or team makes tasks about the same size): 2
 - Tasks burn down only after whole task is done: 2
 - Stories burn down (no tasking) after whole story is done: 2
 - All team members know team's historic Velocity: 1
 - Team commits to sprint backlog at or below Velocity:1

Assessment 8: Retrospection

- As a team, we review our processes, so we can sustainably improve productivity.
- Acceptance Tests (sum):
 - Team conducts retrospectives at least every 2 months: 2
 - Team conducts retrospectives after each Sprint: 2
 - Team limits retrospective participation to team and SM. Team optionally invites PO and others or uninvites SM: 2
 - Team uses sticky-notes/other tools to ensure all members participate and tracks followup: 2
 - Team puts top process improvement in the backlog for next sprint with acceptance tests: 2

Assessment 9: ScrumMaster

- As a team, the ScrumMaster competently enforces process, removes impediments, and provides transparency, so we can focus well.
- Acceptance Tests (sum):
 - SM understands Scrum and agile concepts deeply: 2
 - SM performs no tasks in the Sprint: 1
 - SM enforces rules established by the team: 1
 - SM sees impediments early, and handles for the team: 2
 - SM maintains and uses a prioritized impediments list: 1
 - SM makes team's progress transparent to outsiders: 2
 - SM communicates well with team, other teams, managers, stakeholders and PO: 1

Assessment 10: Team

- As a team, we work together effectively to releasing our software, so we can get software to users earlier and adapt rapidly.
- Acceptance tests (sum):
 - $3 \leq$ team size without counting SM or PO ≤ 7 : 2
 - Team members volunteer (are not assigned) to tasks: 2
 - At least 2 members can independently finish each task: 2
 - Team collectively commits to Sprint goal and backlog: 1
 - Team collectively fights impediments in-sprint: 1
 - Team reduces technical debt every sprint: 2

Agile Manifesto Principles

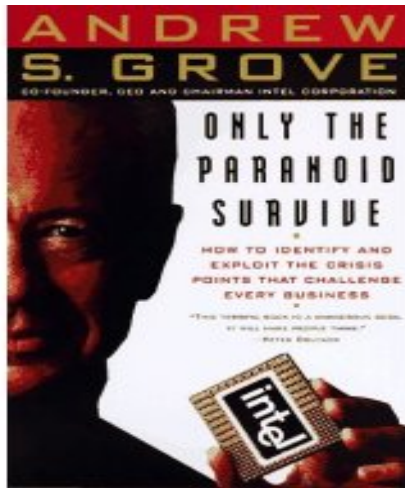
- 1. Our highest priority is to satisfy the customer through early and continuous delivery of customer visible value.
- 2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- 3. Deliver working product frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- 4. Business people and developers must work together daily throughout the project.
- 5. Build projects around motivated individuals. Give them the environment and support they need, and

Agile Manifesto Principles

- 6. The most efficient and effective method of conveying information to and within a delivery team is face-to face conversation.
- 7. Customer visible value is the primary measure of progress.
- 8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- 9. Continuous attention to technical excellence and good design enhances agility.
- 10. Simplicity - The art of maximizing the amount of work not done - is essential.
- 11. The best architectures, requirements, and designs emerge from self-organizing teams.
- 12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.
-

ScrumButt vs. Scrum Design Goal



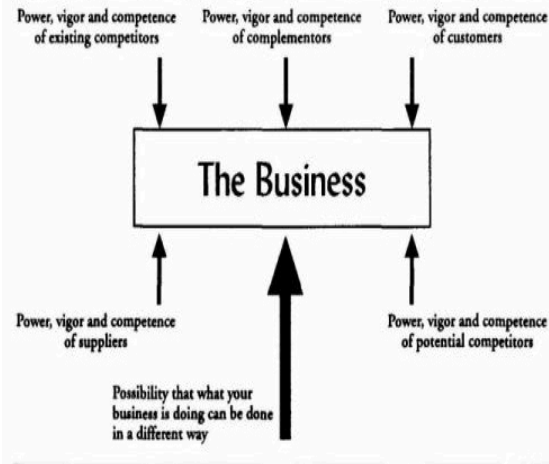


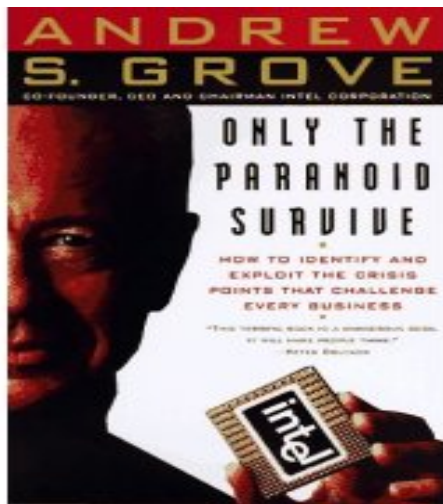
Andy Grove. Only the Paranoid Survive: How to Exploit the Crisis Points That Challenge Every Company: 1st Currency Paperback, 1999.

A "10X" Force

When a change in how some element of one's business is conducted becomes an order of magnitude larger than what that business is accustomed to, then all bets are off. There's wind and then there's a typhoon, there are waves and then there's a tsunami. There are competitive forces and then there are supercompetitive forces. I'll call such a very large change in one of these six forces a "10X" change, suggesting that the force has become ten times what it was just recently. This is illustrated in the following diagram.

Six Forces Diagram—With a "10X" Force



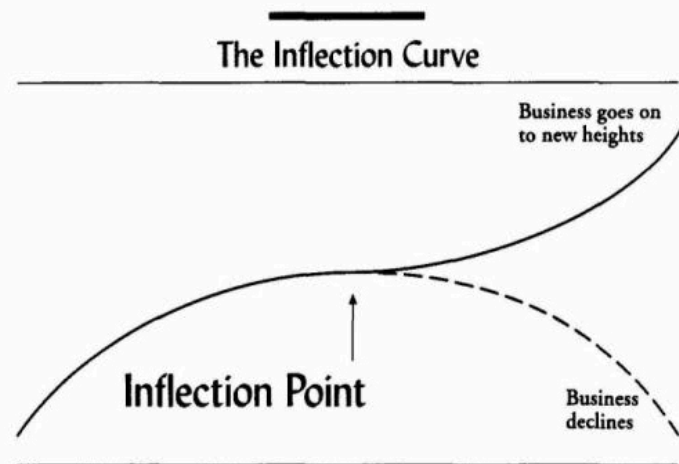


Andy Grove. Only the Paranoid Survive: How to Exploit the Crisis Points That Challenge Every Company: 1st Currency Paperback, 1999.

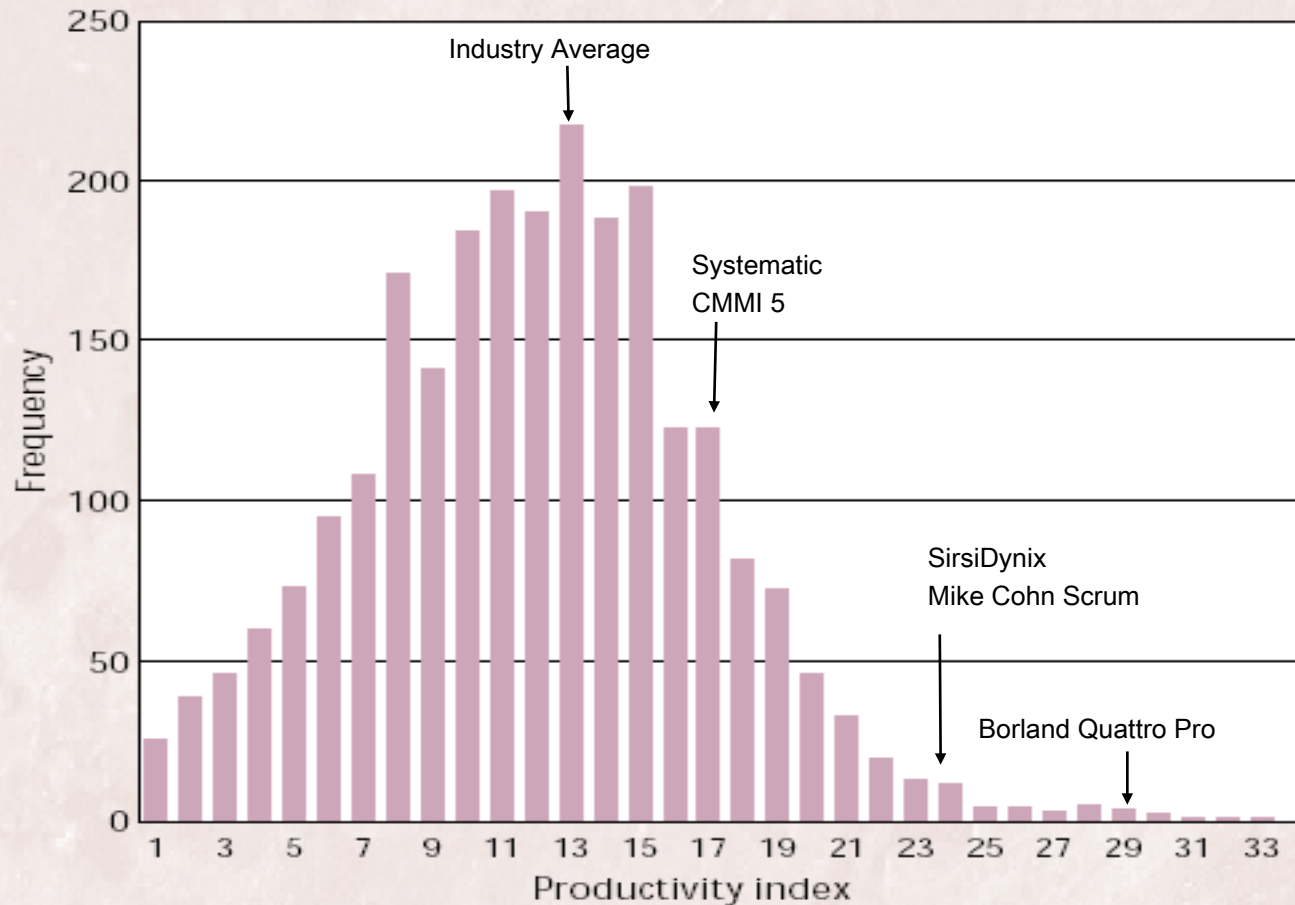
What such a transition does to a business is profound, and how the business manages this transition determines its future. I like to describe this phenomenon as an *inflection point*.

The Strategic Inflection Point

What is an inflection point? Mathematically, we encounter an inflection point when the rate of change of the slope of the curve (referred to as its “second derivative”) changes sign, for instance, going from negative to positive. In physical terms, it’s where a curve changes from convex to concave, or vice versa. As shown in the diagram, it’s the point at which a curve stops curving one way and starts curving the other way.



Design Goal of Scrum - 10x



[Industrial Strength Software: Effective Management Using Measurement](#) by Lawrence H. Putnam and Ware Myers (1997)
Users Stories Applied by Mike Cohn (2004)
Organizational Patterns of Agile Software Development by Coplien and Harrison (2004)
Scrum and CMMI by Sutherland, Jakobsen, and Johnson (2007)
Distributed Scrum (SirsiDynix) by Sutherland, Victorov, and Blount (2006)

If a company can deliver great Scrum, how can they monetize their performance?

- Industry incentives now are for projects to be late.
- Many vendors only make money if the project is late and over budget due to change requests and building functionality the end users do not want.
- CIOs participate in this dysfunctional behavior using their current proposal and contracting process.
- The whole industry could be viewed as driven by bad incentives and faulty practices as 83% of waterfall projects over \$3M fail - see Gartner Group summary of Standish data.

Typical Fixed Price Contract

- Client sends out tender to 3+ potential suppliers. Everything is equally important. Assume total is \$5M.
- All suppliers place a bid of around \$5M.
- One supplier chosen and contract signed.
- Change requests start coming in from day one. All changes are expensive. Project ends up with millions of dollars in change requests.
- After acceptance there still are more work to do because of bugs and some functionality that is not really completed or useful.
- Project cost at end is \$10M - delivered late.

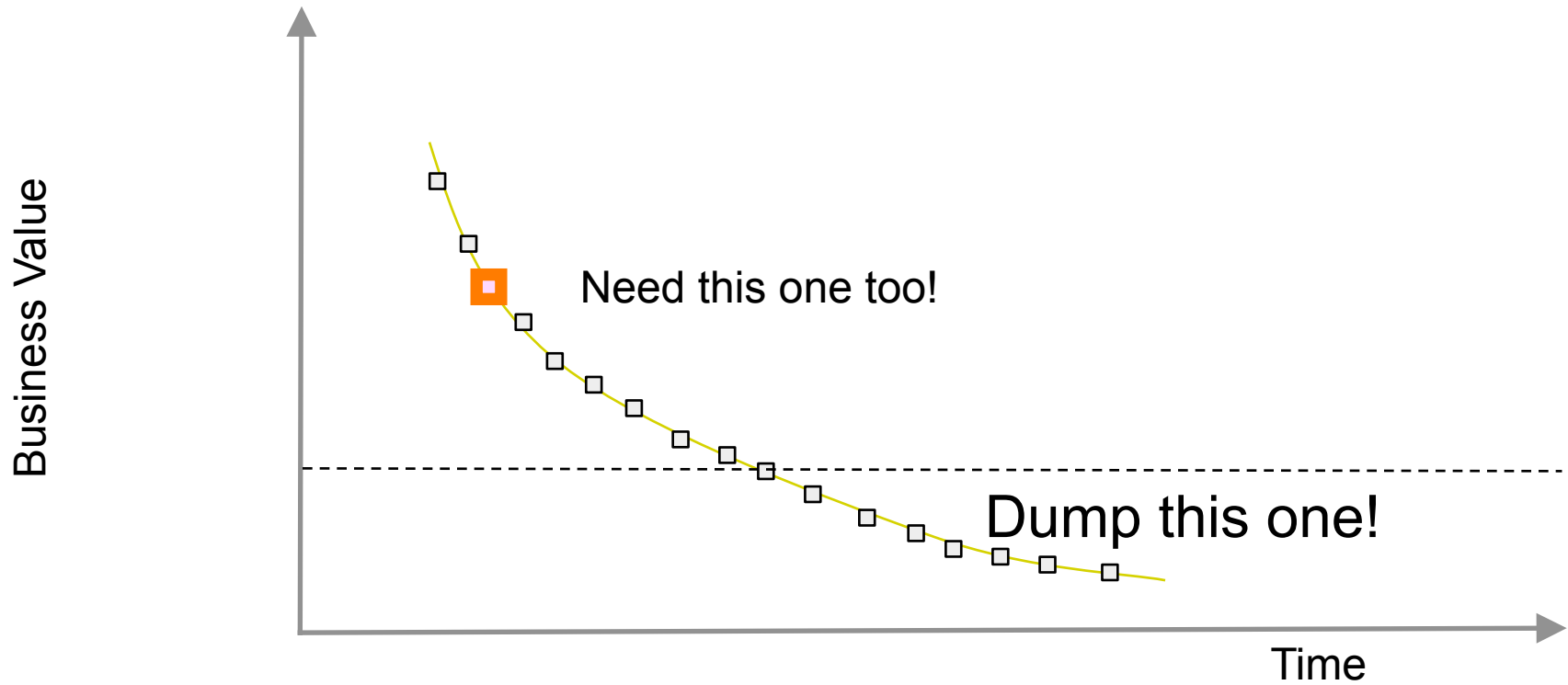
The Alternative - Change for Free

- Use a standard fixed price contract which includes time and materials for changes
- Insert the Change for Free option clause.
 - The customer may execute this option by working with the Scrum Team every Sprint.
- Failure to follow Scrum framework voids this clause and the contract reverts to time and materials.

Change for Free Contract Clause

- The Scrum Product Owner reprioritizes the Product Backlog at the end of each Sprint.
- Changes are included with these rules
 - Changes in priorities are free if total contract work is not changed
 - New features may be added for free at Sprint boundaries if low priority items of equal work are removed from contract.
- Requirements of customer:
 - Features are prioritized by business value and implemented in order of maximum value
 - Users follows project closely and work with the Product Owner to produce a quality Product Backlog
- Each change is tracked with half page contract addendum

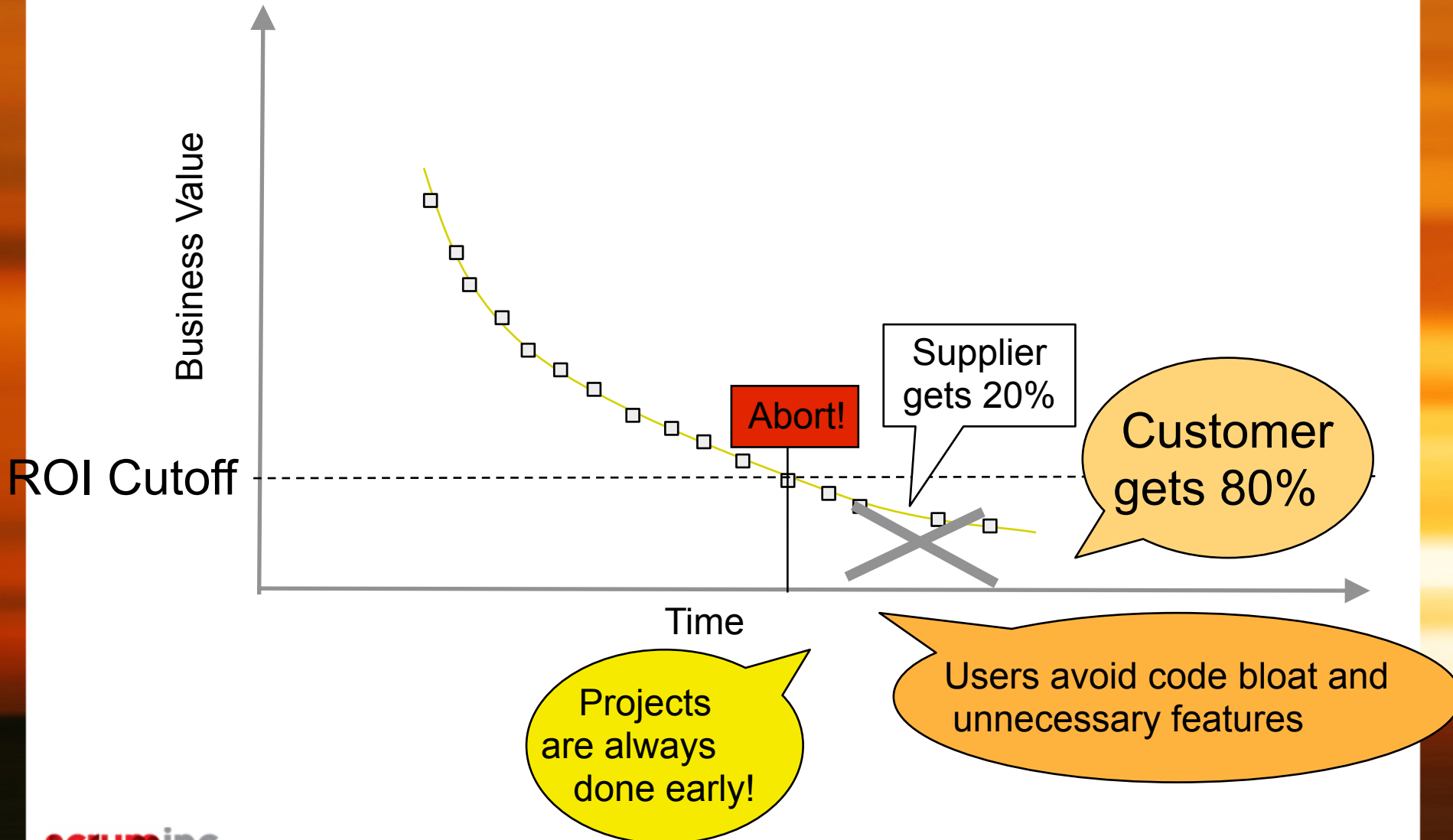
Change for free!



We can do better than Change for Free Money for Nothing!

- Use standard fixed price contract
- Insert Money for Nothing clause.
 - Only operational if customer follows Scrum rules
 - Mutually agreed estimates for all work items
 - Otherwise contract reverts to time and materials
- Customer determines ROI cutoff where implementation of the next feature costs more than the value of the feature.
- Supplier allows termination of contract at any time for 20% of remaining contract value.
- Supplier assumes risk of late delivery of mutually agreed work.

Money for Nothing!



Fixed Price, Fixed Date

Money for Nothing and Change for Free

Contract provisions:

1. Customer involvement allows us to tune the system to the latest known business value.
2. Any requirement that hasn't already been worked on can be swapped out for another of equal value;
3. Priority of requirements can be changed by customer;
4. Customer may request additional releases at any time at prevailing time and material fees;
5. Customer may terminate contract early if value has been satisfied for 20% of remaining unbilled contract value

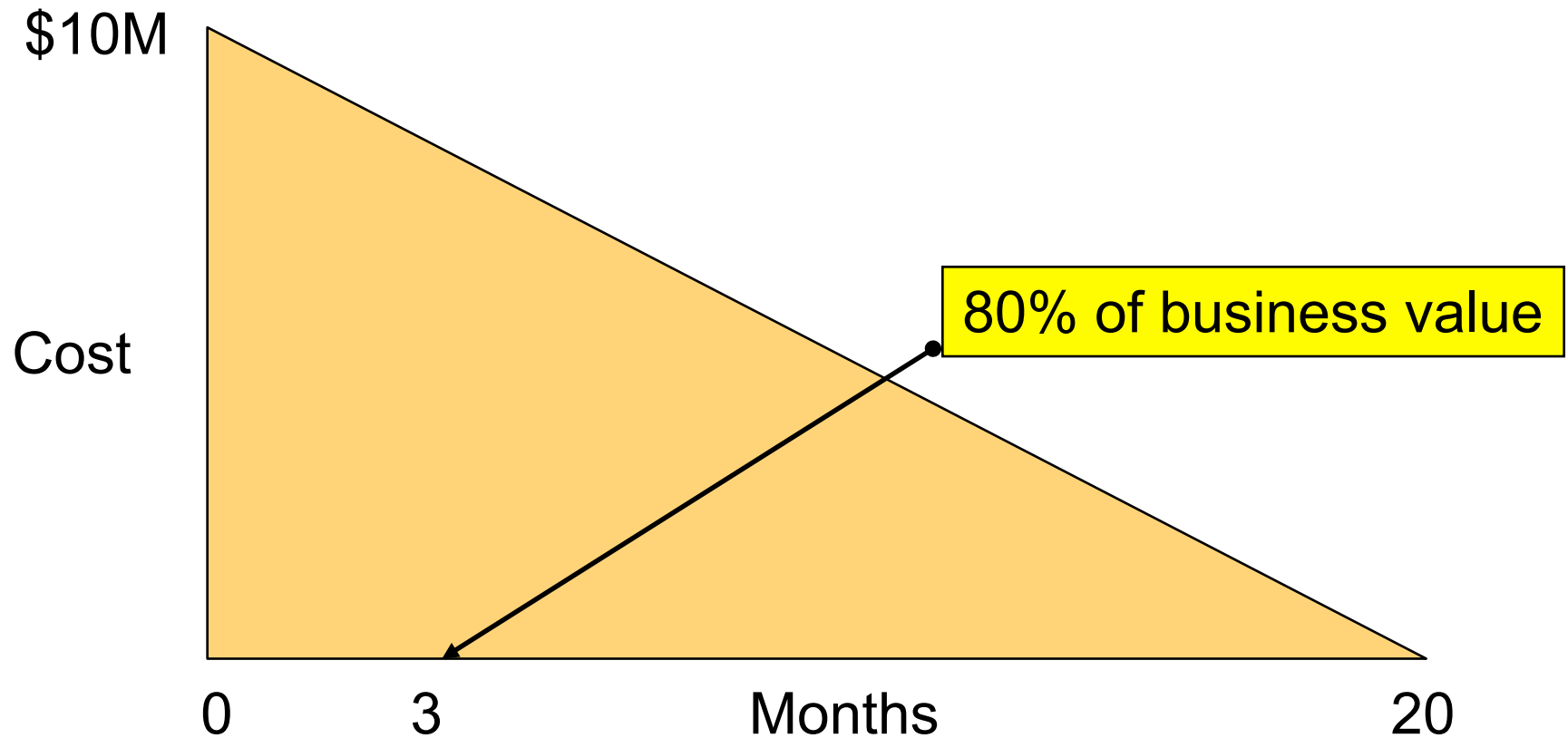
Fixed Resources, Fixed Date

Money for Nothing and Change for Free

Development plan:

1. Product Owner involvement allows us to tune the system to the latest known business value.
2. Any requirement that hasn't already been worked on can be swapped out for another of equal value;
3. Priority of requirements can be changed by Product Owner;
4. Product Owner may request additional releases at any time at prevailing time and material schedules;
5. Product Owner terminates development and releases product as soon as value of next feature is less than cost of not shipping early.

Project Management Software for Construction Company - \$10M



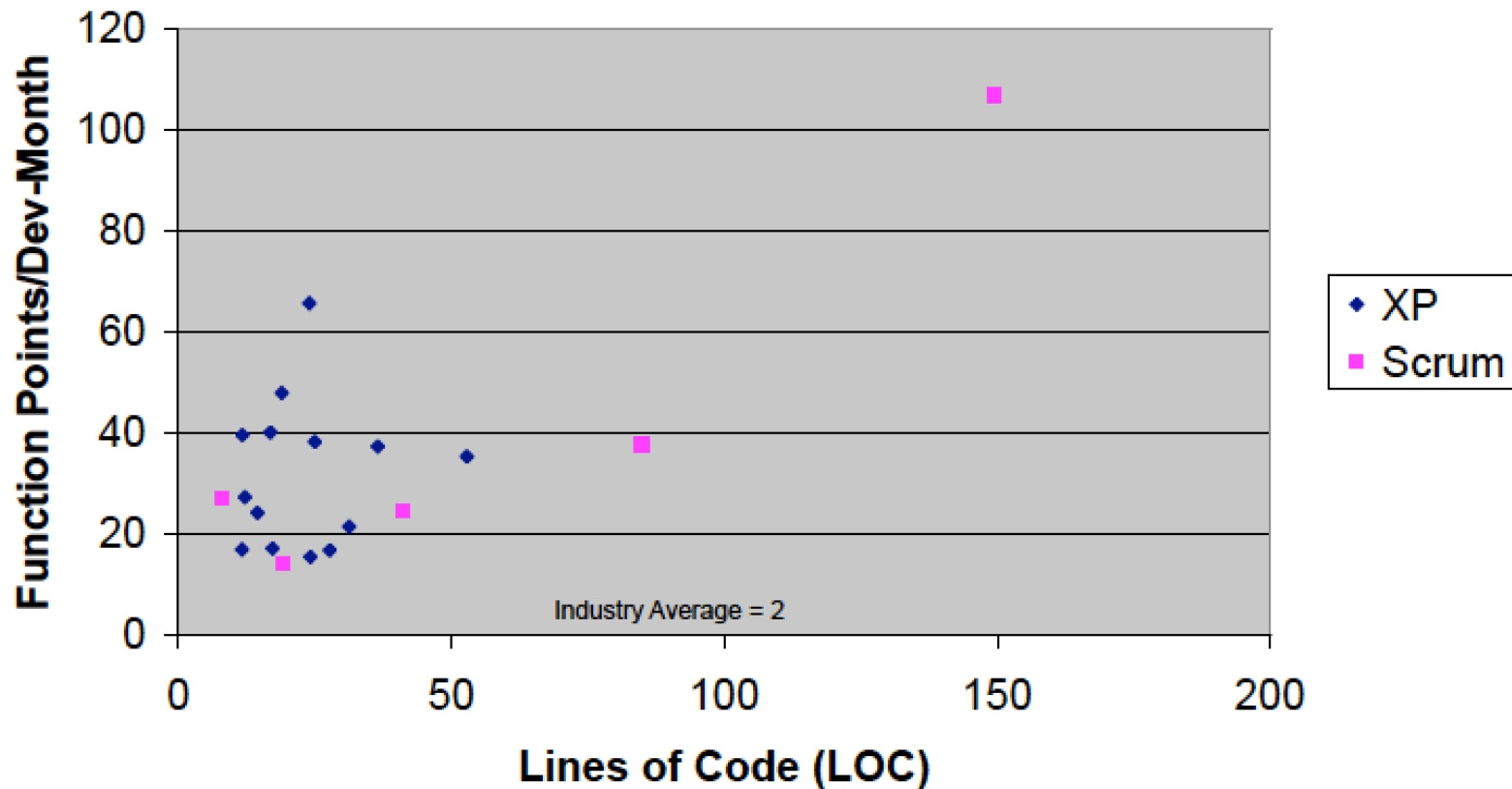
Early Termination

Money for Nothing!

- 15% of \$10M = \$1.5M
- 20% of \$8.5M = \$1.7M
- Total = \$3.2M
- Cost to build = \$1.3M
- Margin 15% → 60%
- Earnings increase by 400%
- Early retirement strategy

Russian projects velocity data suggests high velocity is not an accident

Exigen Services Recent C# Projects



Exigen Services

- Over 2000 developers
- Agile division in St. Petersburg has virtually all hyperproductive teams
- “Money for Nothing” is strategic imperative to capture value of high velocity production
- Requires major training and upgrade of procedures for engaging with customers in management, marketing, and deployment groups.
- ***Disruptive technology*** for dismantling worldwide waterfall market of late projects over budget with unnecessary features, poor quality,

Results of Customer Research

- Set up CIO dinners in London and New York for 50 people.
- 1/3 of attendees ready to start contracting using “Money for Nothing” strategy.
- 1/3 ready to start talking about using this strategy and wanted sales team engagement.
- 1/3 said their organizations were too dysfunctional to execute this strategy
 - could not get good product backlog
 - could not prioritize features by value
 - lack of trust between management, development, and vendors

CIO Requests

- CIOs want to know velocity of vendor teams before committing to long term project
- Early short term engagement to develop product backlog and validate development team velocity was viewed as desirable.
- Long term contract negotiations based on real velocity of early teams.

Example: Flex-Agility 2.0

- Flex-Agility 2.0 is a premium product
 - It is not the lowest cost way or even the quickest way to conduct a project with Exigen
 - It does not fit all contracts
- It is a way to guarantee a delivery and still have the option for a high degree of change
- It is not for all customers
 - We may say “no” and work on T & M
- T & M is low risk so standard T & M is OK
- Flex-Agility 2.0 is more shared risk and so commands a premium
- We deliver highest business value first and so early termination with value is a real and desired outcome

Flex-Agility 2.0 Value Proposition

- Larger projects (>500K) where customer wants guarantees or shared risk with Agile flexibility
- Buy vs. Build
 - Certainty of Buy but with bespoke Build
- Guaranteed velocity and estimates
 - Commercial penalties for underachieving velocity
- Best endeavors to correct problems
- Business value rather than head count tracking and billing
- Option of closing early should enough business value be achieved – this is “Money for Nothing”
- Option of adding new requirements into scope during project by replacing with lower priority requirements of equivalent “size” – this is “Change for Free”

Exigen Services White Paper Outlining Next-Generation Outsourcing Engagement Model

San Francisco, July 8, 2008 – Exigen® Services, the leading next-generation application outsourcing provider, today announced the availability of a free white paper that details how fixed price Agile in a distributed outsourcing environment works. The white paper, titled “Unlimited Change for A Fixed Price: the Next Generation of Outsourcing Contracts” provides the framework for establishing a truly collaborative model that further aligns IT and the business with their outsourcing provider. Download the paper here: www.flex-agility.com.

Challenges

- Getting teams to deliver is easy for Exigen
- Managing customer to get prioritized backlog is hard
 - Backlog is high level and not detailed enough for developers to estimate well
- Next steps - train all engagement managers to be good product owners
- Raise maturity level of estimation process
 - Build historical database to assist in estimation

Recommendations

- ScrumButt

- Stick to time and materials body-shopping with low margins
- Work hard for the rest of your life



- Hyperperforming teams

- Monetize your performance
- For five times the velocity, get five times the margins
- Use “Money for Nothing and Change for Free” strategy



- Make the world a better place by altering the fundamental structure of the IT industry

- Implement the design goal of Scrum, bring all projects in early, disrupt waterfall competitors, and execute the early retirement plan!

Questions?

