Roger Sessions

IT Complexity Crisis; Danger and Opportunity

ObjectWatch

Agenda

- The Failure of Failure
- Another Approach to Failure Analysis
- Measuring Complexity
- A Practical Example
- SIP: A Methodology for Reducing Complexity
- Summary
- Call to Action

Two Studies

- 2009 Chaos report published by the Standish Group

- The Impact of Size and Volatility on IT Project Performance by Chris Sauer, Andrew Gemino, and Blaize Horner Reich, Comms of the ACM Nov 07

2002 Data on IT Success

- Standish:
  - Successful: 28%
  - Challenged: 51%
  - Failed: 20%
- Sauer
  - Good or Star: 68%
  - Challenged: 23%
  - Failed: 9%

Who is right?

Four Problems with Studies

- Poor definitions
- Self selected respondents
- No reporting of statistical significance
- Wrong analysis

The Wrong Analysis

P1	$100K	Succeed
P2	$100K	Succeed
P3	$250K	Succeed
P4	$300K	Succeed
P5	$5M	Fail

According to Standish/Sauer, this is a 75% success rate.
(4 out of 5 projects successful)

But based on IT budget, this is a 13% success rate.
($750K out of 5,750K delivered value.

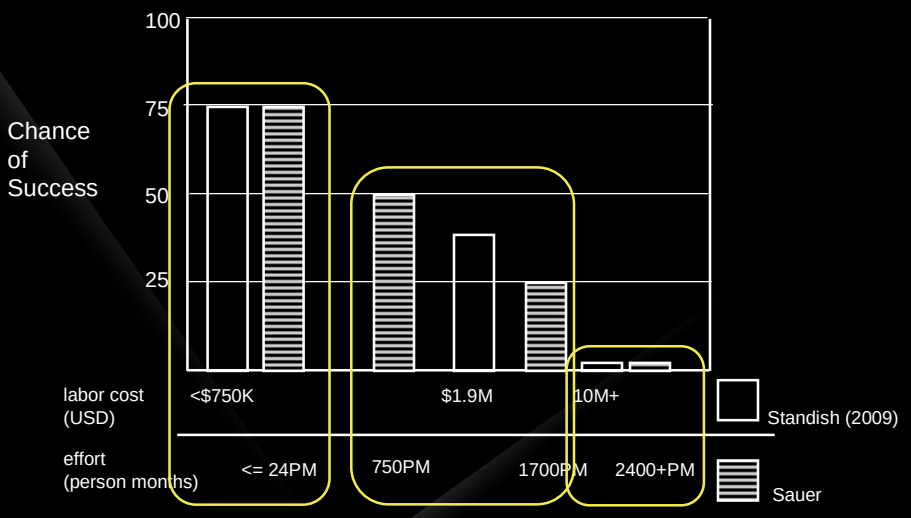
Which is right?

BUT

What if all projects are equally likely to fail?

Then the Standish/Sauer numbers make more sense.

Comparing Both Studies



Four Conclusions

Failure rates go up exponentially with size.

If your project is more than $2M, it will probably not be successful.

If your project is more than $10M, it will definitely not be successful.

It makes no difference what methodology you use.

Agenda

- The Failure of Failure
- Another Approach to Failure Analysis
- Measuring Complexity
- A Practical Example
- SIP: A Methodology for Reducing Complexity
- Summary
- Call to Action

The Cost of Failure

2.75 %	GDP Spent on hardware, software, services
66 %	Federal IT dollars in “at risk” projects
65 %	“at risk” dollars will be lost
7.5	indirect costs for every dollar direct costs.

.0275 X .66 X .66 X 7.5 = .089 = Failure factor

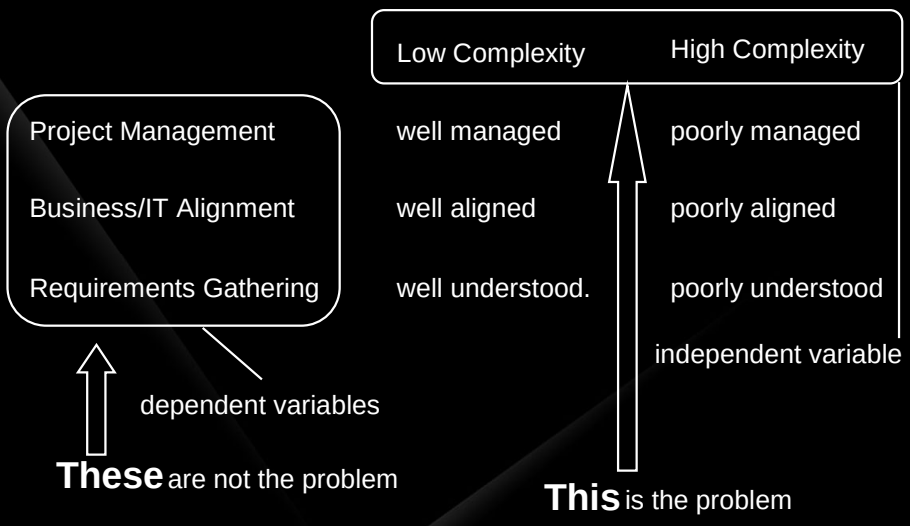
Example (UK)

GDP of UK = $2,260 B

.089 X $2,260 B = $200 B/year

= $548 M/day

Who is the enemy?



Agenda

- The Failure of Failure
- Another Approach to Failure Analysis
- Measuring Complexity
- A Practical Example
- SIP: A Methodology for Reducing Complexity
- Summary
- Call to Action

Glass's Law

For every 25 percent increase in problem complexity, there is a 100 percent increase in solution complexity.

- Facts and Fallacies of Software Engineering by Robert Glass

Glass's Law Applied to SOAs

For every 25 percent increase in the business functionality in a service, there is a 100 percent increase in the complexity of that service.

For every 25 percent increase in the number of connections in a service, there is a 100 percent increase in the complexity of that service.

SCU = Standard Complexity Unit

The amount of complexity in an average atomic business function implemented in an idealized green field environment with no internal or external interactions.

Examples of atomic business functions: process-check, hire-employee, remove-from-inventory

Bird's Formula

Glass's Law

1.25^{Fx} functionality = 2^{Cx} X complexity

Bird's Formula

A system with bf business functions is

$10^{\log(Cx) \times \log(bf) / \log(Fx)}$

times more complex than a system of 1 business function.

Simplifying Bird's Formula

$10^{\log(Cx) \times \log(bf) / \log(Fx)}$

is with constants replaced

$10^{\log(2) \times \log(bf) / \log(1.25)}$

Equivalent to

$10^{.30103 \times \log(bf) / .09691}$

Equivalent to

$10^{3.10 \log(bf)}$

Glass's Constant

Sessions's Formula for Service Complexity

A system with bf business functions and cn connections has a complexity value (in SCUs) of

$10^{\log(Cx) \times \log(bf) / \log(Fx)} + 10^{\log(Cx) \times \log(cn) / \log(Fx)}$

Equivalent to

$10^{3.10 \log(bf)} + 10^{3.10 \log(cn)}$

Sessions's Formula for SOA Complexity

An SOA with m services has a total complexity (in SCUs) of

$$\sum_{i=1}^{m} 10^{3.10\ \log(bfi)} + 10^{3.10\ \log(cni)}$$

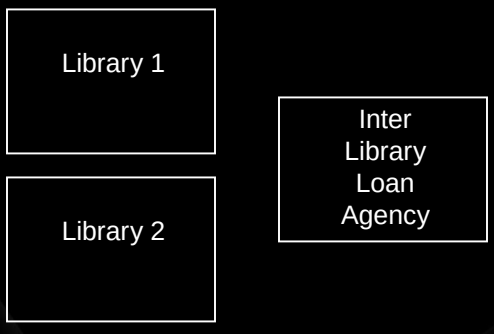
Agenda

- The Failure of Failure
- Another Approach to Failure Analysis
- Measuring Complexity
- A Practical Example
- SIP: A Methodology for Reducing Complexity
- Summary
- Call to Action

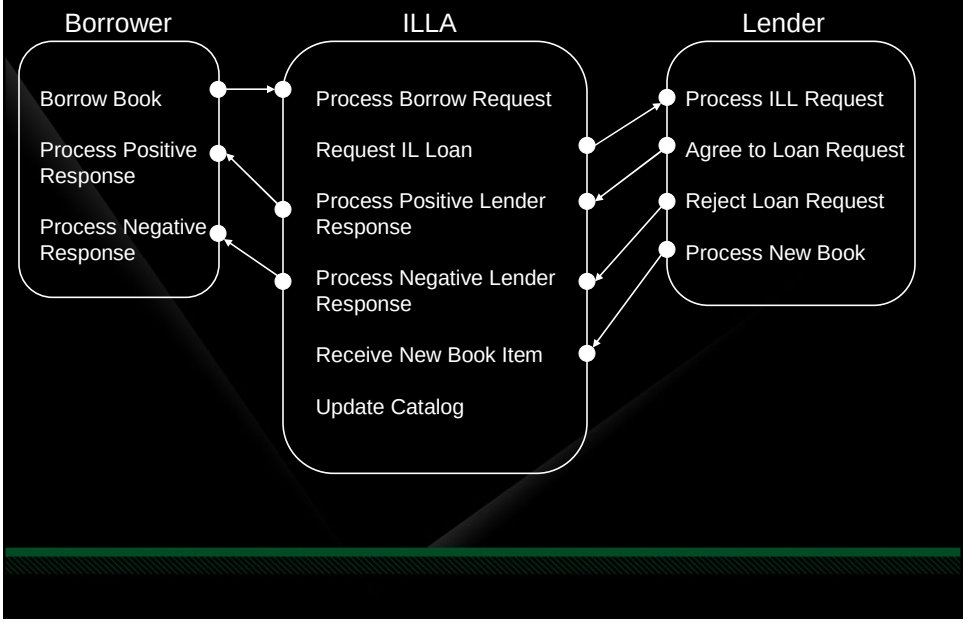
Practical Problem

- Say architecture A and B both solve a business problem
- Say A is twice as complex as B
- Then A will cost twice as much, work half as well, and be twice as likely to fail as B
- BUT ... how do we know which is more complex?

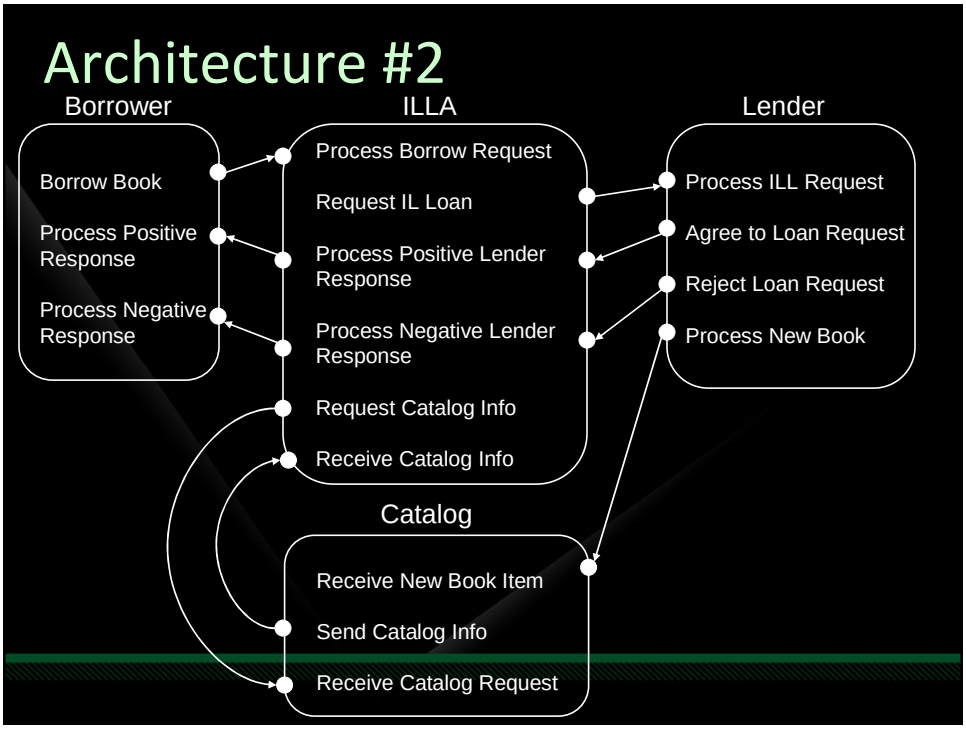
Example: Interlibrary Loan

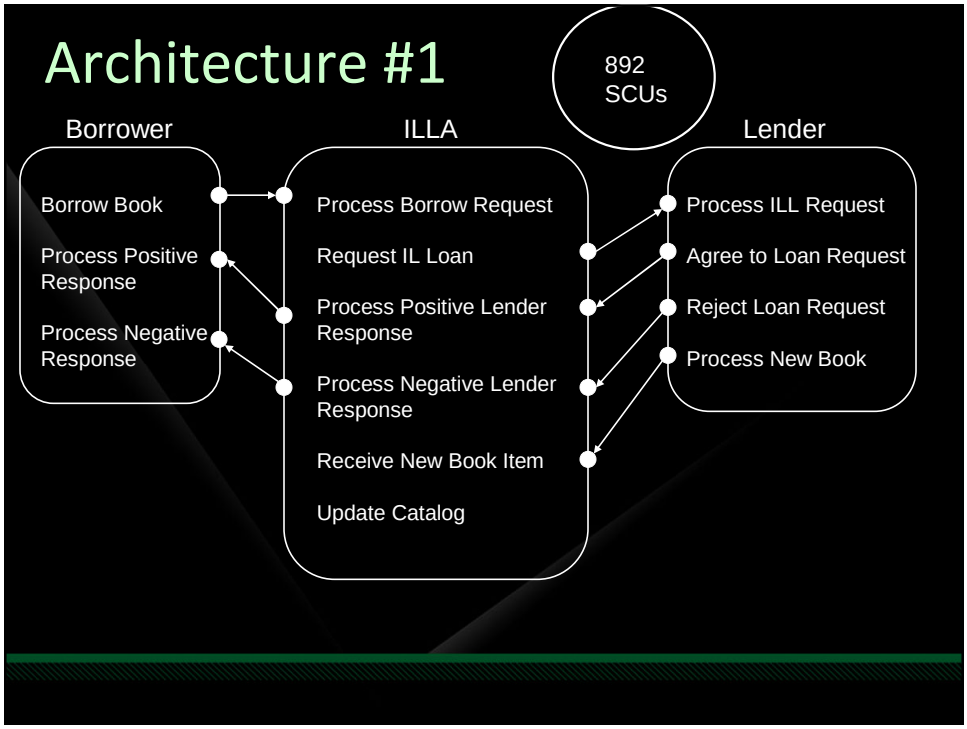


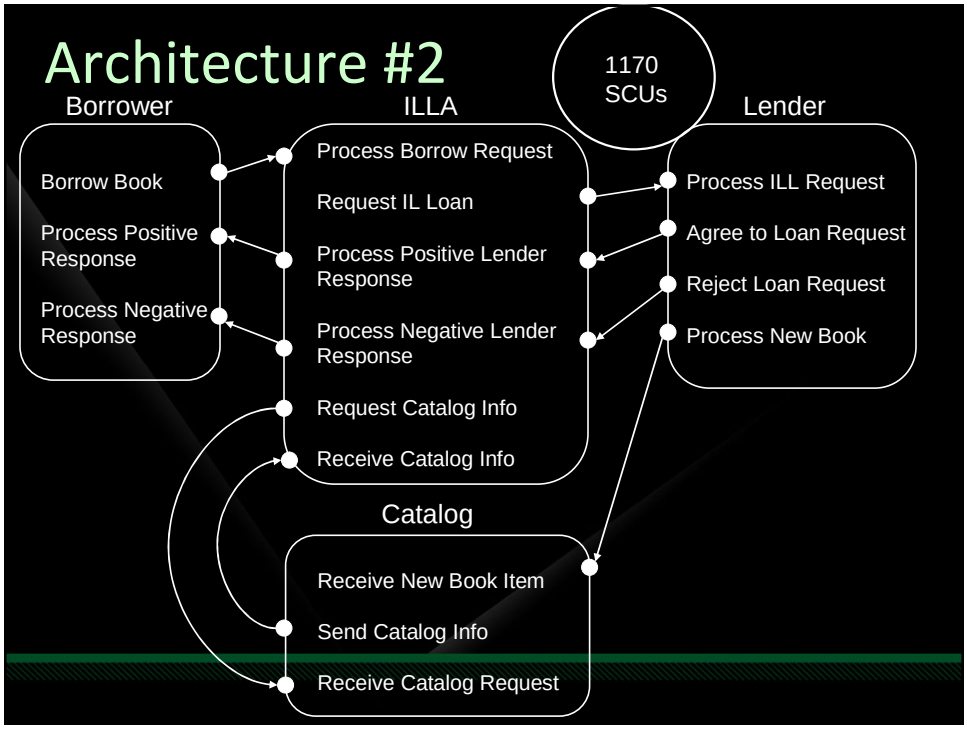
Architecture #1



Architecture #2







Two Architectural Solutions

- Solution 1: 892 SCUs
- Solution 2: 1170 SCUs
- Solution 2 will cost 30% more than Solution 1 and be 30% more likely to fail.

The Solution Space

Problem P has n solutions: {s1, s2, s3, ... sn}

Every solution si has a complexity measure C(si)

We want to find the solution sx that has the lowest complexity measure C(sx)

If P contains 20 business functions, n > 50 trillion.

We need a process to drive us to sx.

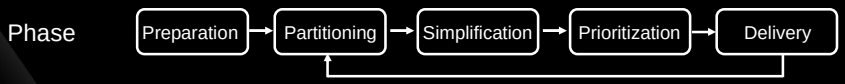
Agenda

- The Failure of Failure
- Another Approach to Failure Analysis
- Measuring Complexity
- A Practical Example
- SIP: A Methodology for Reducing Complexity
- Summary
- Call to Action

SIP

Simple Iterative Partitions

SIP Process



The Process - Decomposition

Interlibrary Loan

Borrower
- Ask To Borrow
- Process Positive Response
- Process Negative Response

Lender
- Process Loan Request
- Agree To Loan
- Reject to Loan
- Process New Book
- Update-Catalog

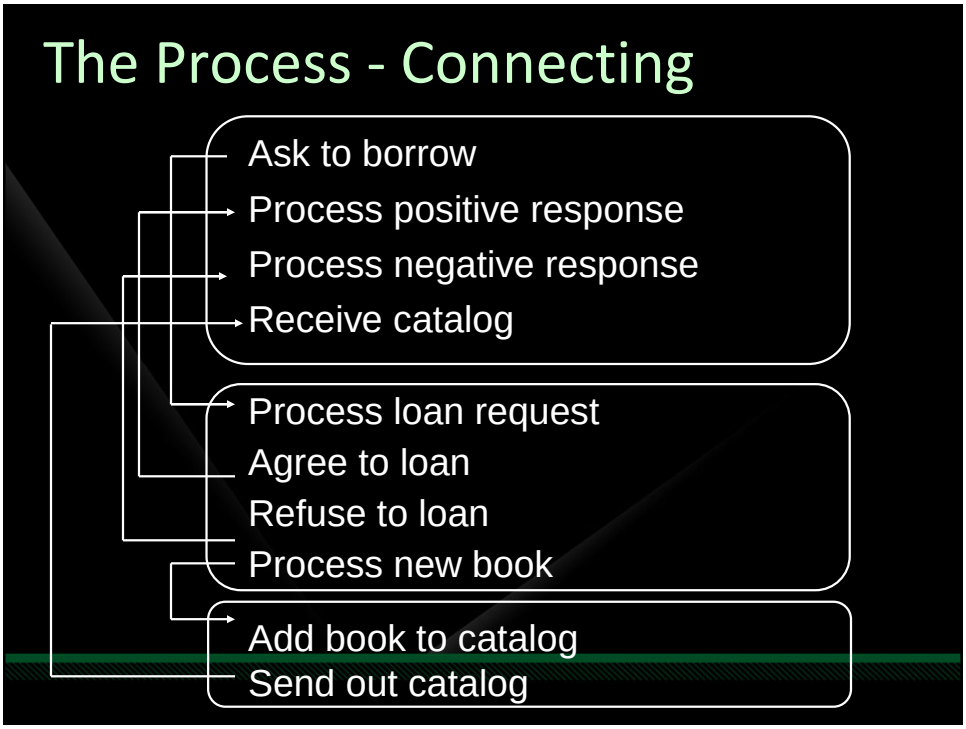
Partitioning and Synergistic Analysis

Ask to borrow	A
Process loan request	B
Agree to loan	B
Process positive response	A
Refuse to loan	B
Process negative response	A
Process new book	B
Add book to catalog	C

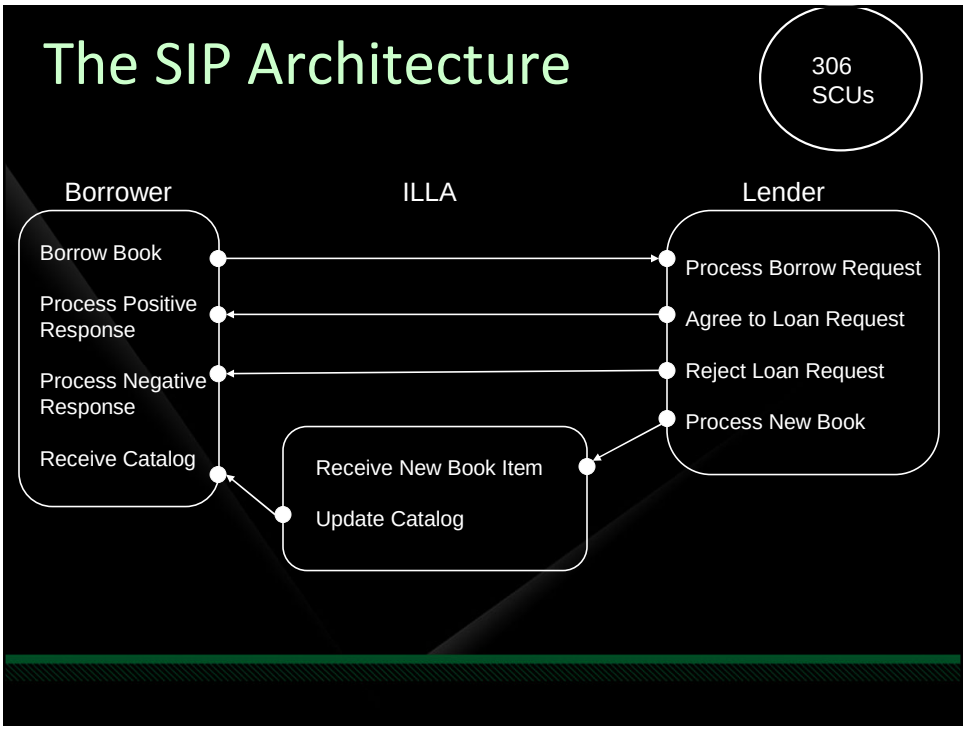
The Process - Subsetting

Ask to borrow	A
Process positive response	A
Process negative response	A
Receive catalog	A
Process loan request	B
Agree to loan	B
Refuse to loan	B
Process new book	B
Add book to catalog	C

The Process - Connecting



The SIP Architecture



Stepping Back

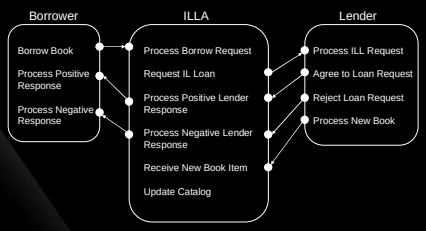
- Standard Architecture 1: 892 SCUs
- Standard Architecture 2: 1170 SCUs
- SIP Architecture: 314 SCUs

The SIP Generated Solution is 35% of the cost of the *least expensive* non-SIP solution.

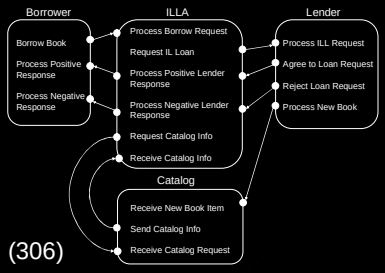
The best non-SIP Solution is almost 3 times more *likely to fail* as the SIP solution.

Side by Side

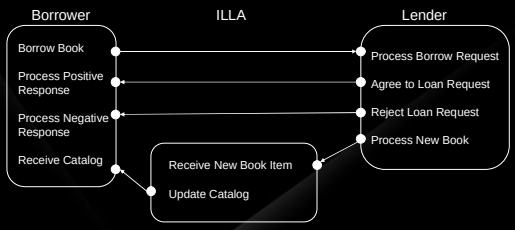
Architecture 1 (892)



Architecture 2 (1170)



SIP Architecture (306)



Goals of SIP

- A process to drive the simplest possible system design that solves a given business problem
- Based on mathematical models of complexity, partitioning, and set theory.
- Reproducible and verifiable.

Agenda

- The Failure of Failure
- Another Approach to Failure Analysis
- Measuring Complexity
- A Practical Example
- SIP: A Methodology for Reducing Complexity
- Summary
- Call to Action

Presentation Summary

- Project success rates is meaningless.
- Budget success rates are important.
- There appears a strong correlation between complexity and success.
- We can measure complexity.
- We can minimize it.
- We can save a lot of money.

It's simple to make things complex.

It's complex to make things simple.

But simple things are the only things that work.

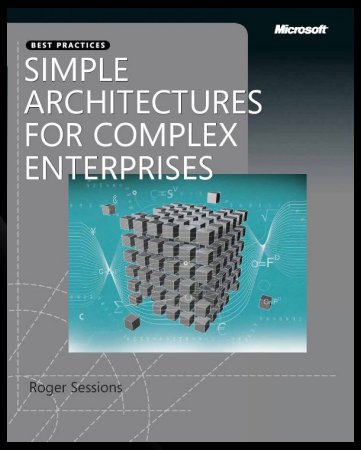
Agenda

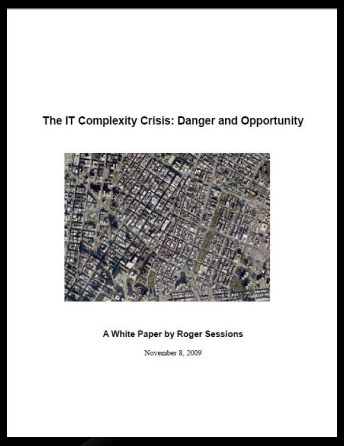
- The Failure of Failure
- Another Approach to Failure Analysis
- Measuring Complexity
- A Practical Example
- SIP: A Methodology for Reducing Complexity
- Summary
- Call to Action

Call To Action

- A major opportunity to improve the world.
  - Financial Benefits
  - Efficiency Benefits
  - Social Benefits
- We need to work together to investigate this problem.
  - Research
  - Better analytic tools
  - Standardized complexity management processes
- We need a consortium to make this happen.

For more information





twitter: @rsessions
email: roger@objectwatch.com
www.objectwatch.com