From Patterns: Eastward to Lean, Westward to true objects

James O. Coplien
Scrum Training Institute
Gertrud & Cope
cope@gertrudandcope.com

基本的なパターンのプロセスクリストファーAlexanderによって作成されたコミュニティのチームワーク、即座にフィードバックし、継続的な改善に基づいています。しかし、プロセスが重要です：永遠のフォームの構図は、ソウルフルな全体のフォームを達成するだけの方法です。パターンコミュニティの創設者のコミュニティ、更新、これらの理想は、断片的なわれわれの作ったものをフォームと美しさの組成を考えていた。

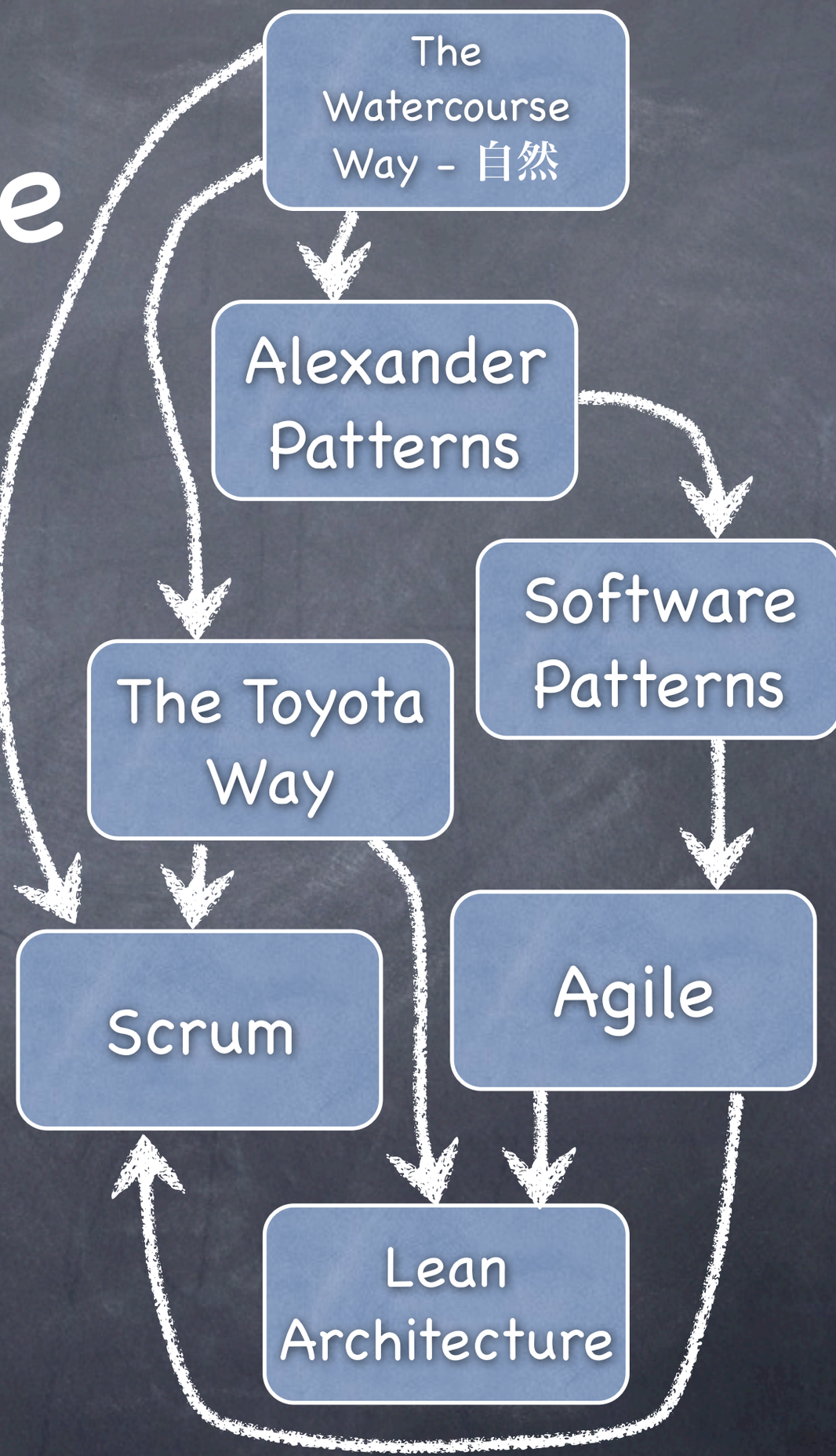
過去10年間に我々はコミュニティの共通のパターンを改善している必要があります。彼らは、マイクロ始めたなアーキテクチャと、私たちは本当のWholesになるまでに成長していいのになあ。私たちは洗練された、公開のパターンの多くの世代を望む。私たちのパターンを当社のシステムの機能の重要されているフォームを表示する必要があります。パターンは、これまで出ていないし、彼らが十分なソフトウェアの複雑な芸術、建築のニーズに対応するための強力ではありません。

しかし、人々が新しいソフトウェアプラクティスに彼らとは、キーパターン原則を撮影している。アジャイル今日は非常には人気があるものアジャイル実際にリーンと呼ばれ、ほとんど。スクラムの詳細リーンより機敏されます。米国人作家"と呼ばれるトヨタウェイ""リーン"は、1991年に出版した。リーンのコミュニティチームワーク、即座にフィードバック、カイゼンされます。断片的な成長と現地適応の強いアジャイルな値です。は、パターンのプロセス側になります。トリグベReenskaugからDCIのアーキテクチャ製品の顧客認知からの概念によって形を作成します。 DCIの発展の基礎欧米のオブジェクトの基盤を指向プログラミングのようなものです。しかし、DCIのオブジェクトについて、いくつかの誤解を訂正して、オブジェクトのいくつかの障害を4年間に残留している指向プログラミングを削除します。パターンと同様に、DCIのシステム設計の問題に焦点を当てます。これは、ドメインオブジェクトの不朽のフォームに根差しているダイナミックな構図のテクニックを使用します。は、パターンの製品側になります。

元のソフトウェアパターンのビジョンが失われている。スクラム私優れたプロセスを与えることができます。スクラムトヨタウェイから来ている。建物のタイムレス道道徳経から来ている。 2人の日本人のルーツ。リーンやアジャイルなアーキテクチャにより、良質の製品を与える可能性があります。良い製品を、人間の快適性、そしてシステム思考の経験から来る。これらの具体的な、パターン原則の実用的なアプリケーションは、おそらくパターンコミュニティのためのインスピレーションとガイダンスを提供することができます。

Outline

1. Another Japanese link: Lean
2. Scrum & Agile
3. Back to Geometry: Lean Architecture and DCI
4. What are patterns — really?



1. Another Japanese link: Lean
2. Scrum & Agile
3. Back to Geometry: Lean Architecture and DCI
4. What are patterns — really?

1. Another Japanese link: Lean

- Shares 自然 ancestry with Alexander's Patterns
- Lean has inspired more software practices than Patterns have
- Examples:
  - Scrum (process)
  - Lean Architecture (product)
  - Six Sigma (usually badly practiced)
  - Kanban (represents a shallow misunderstanding of Lean)

Another Japanese link: Lean
Shares tzu-jan ancestry with Alexander's patterns
Lean has inspired more software values than Patterns have
Examples:
  Scrum (process)
  Lean Architecture (product)
  Six Sigma (usually badly practiced)
  Kanban (represents a shallow misunderstanding of Lean)

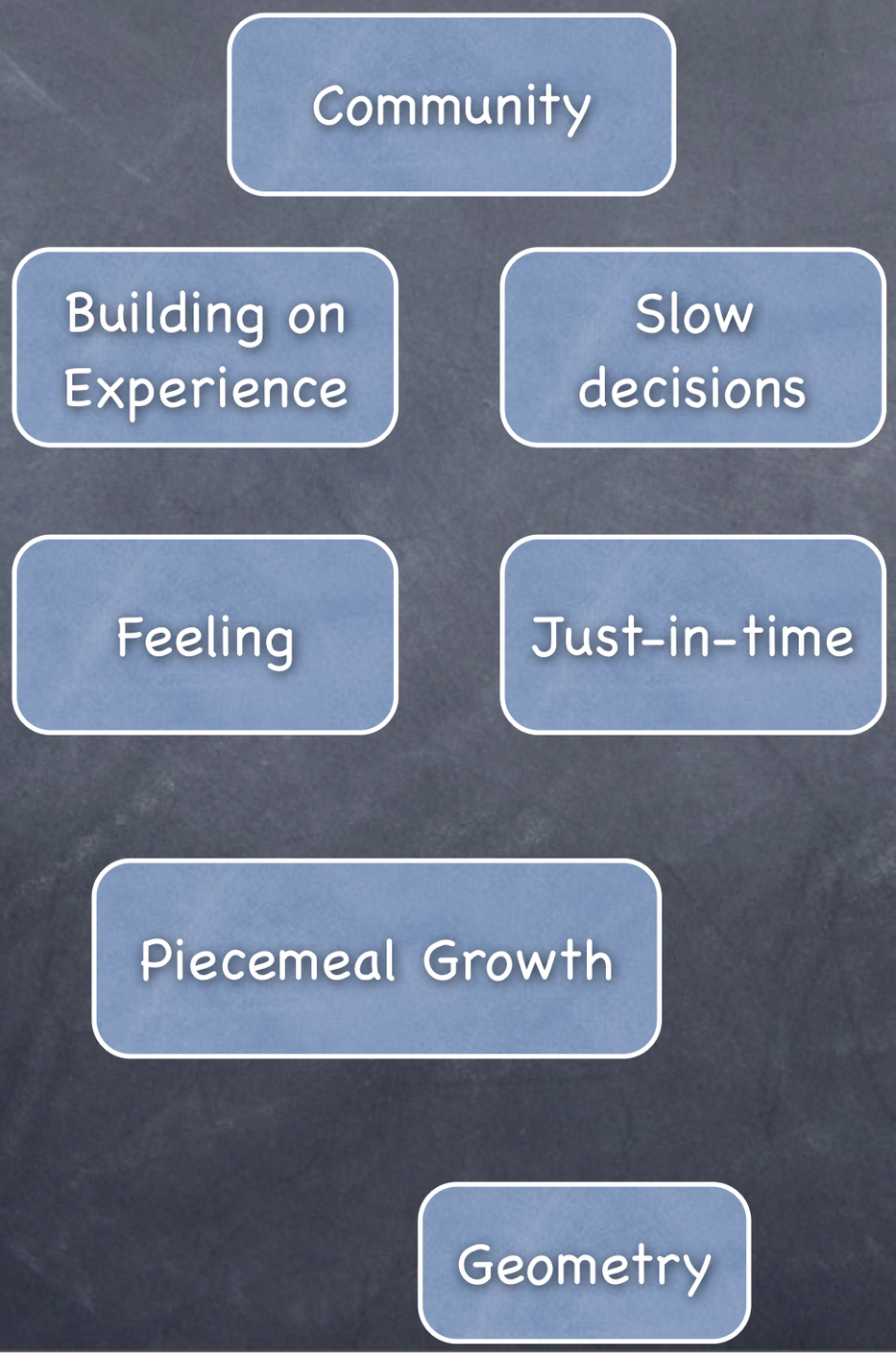
East to West: Toyota USA Motor Manufacturing

1. As an American company, contribute to the economic growth of the **community** and the United States.
2. As an independent company, contribute to the **stability and well-being of team members**.
3. As a Toyota group company, contribute to the overall growth of Toyota by adding **value to our customers**.

G&C

East to West: Toyota USA Motor Manufacturing
1. As an American company, contribute to the economic growth of the **community** and the United States.
2. As an independent company, contribute to the **stability and well-being of team members**.
3. As a Toyota group company, contribute to the overall growth of Toyota by adding **value to our customers**.

Alexander's Pattern Values



Community
Building on Experience
Feeling
Piecemeal growth

Slow Decisions
Just-in-time

Geometry

Alexander's Pattern Values

Community

Building on Experience

Slow decisions

Feeling

Just-in-time

Piecemeal Growth

Geometry

Community
Building on Experience
Feeling
Piecemeal growth

Slow Decisions
Just-in-time

Geometry

The Toyota Way Values

Invest in the Community

根回し

Building on Experience

Just-in-Time

Reflection

Continuous Improvement

Invest in the Community

Nemawashi

Building on Experience

Just-in-time

Reflection

Continuous Improvement

The Toyota Way Values

Invest in the Community

根回し

Building on Experience

Just-in-Time

Reflection

Continuous Improvement

Invest in the Community

Nemawashi

Building on Experience

Just-in-time

Reflection

Continuous Improvement

Comparing Values

Serving the Community

Invest in the Community

Building on Experience

Slow decisions

根回し

Building on Experience

Feeling

Cash flow economics

Just-in-time

Reflection

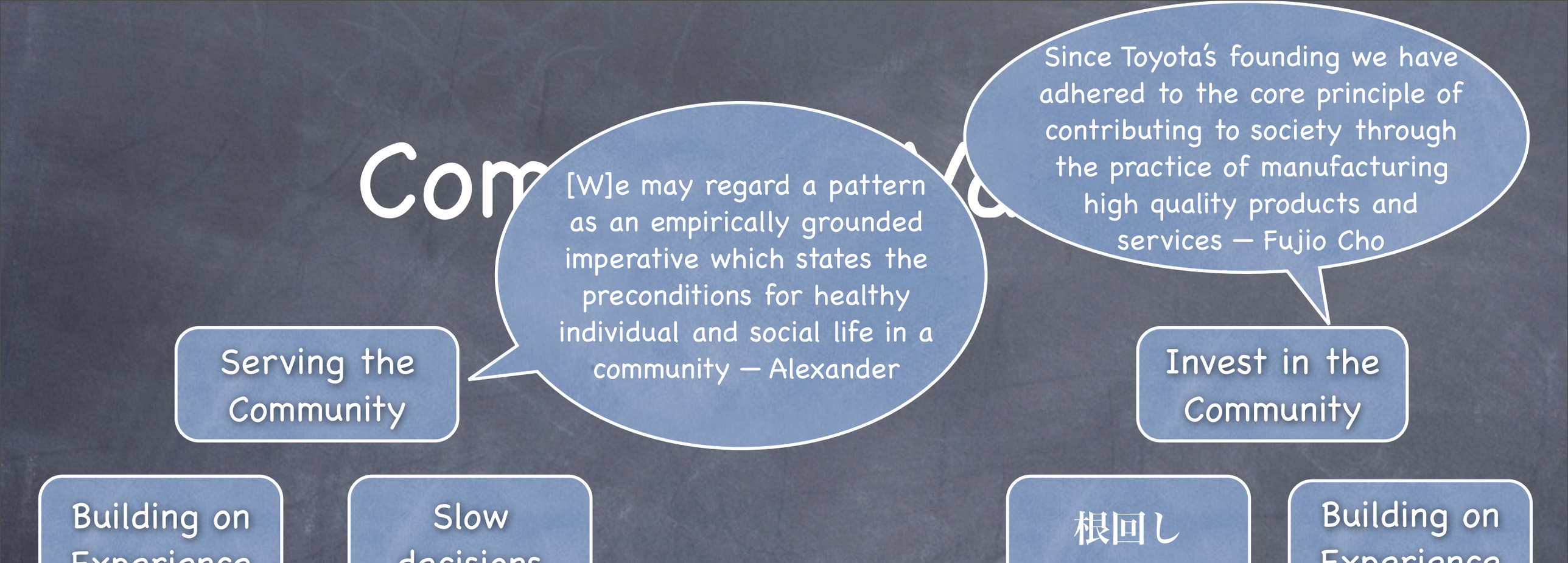
Continuous Repair, Piecemeal Growth

Continuous Improvement

Geometry

Serving the community / Invest in the community

- “....we may regard a pattern as an empirically grounded imperative which states the preconditions for healthy individual and social life in a community.” Christopher Alexander, The Oregon Experiment, Chapter 4
- “The purpose is... to help society and to help the community, and to contribute back to the community that we’re fortunate enough to do business in.” Jim Press, COO of Toyota Motor Sales in North America, Toyota Way, p. 72
- “Since Toyota’s founding we have adhered to the core principle of contributing to society through the practice of manufacturing high quality products and services” — Fujio Cho, in Liker, The Toyota Way, p. 35



Serving the community / Invest in the community

- "....we may regard a pattern as an empirically grounded imperative which states the preconditions for healthy individual and social life in a community." Christopher Alexander, The Oregon Experiment, Chapter 4
- "The purpose is... to help society and to help the community, and to contribute back to the community that we're fortunate enough to do business in." Jim Press, COO of Toyota Motor Sales in North America, Toyota Way, p. 72
- "Since Toyota's founding we have adhered to the core principle of contributing to society through the practice of manufacturing high quality products and services" — Fujio Cho, in Liker, The Toyota Way, p. 35

Comparing Values

Serving the Community

Invest in the Community

Building on Experience

Slow decisions

根回し

Building on Experience

Feeling

Cash flow economics

Just-in-time

Reflection

Continuous Repair, Piecemeal Growth

Continuous Improvement

Geometry

Serving the community / Invest in the community

- “....we may regard a pattern as an empirically grounded imperative which states the preconditions for healthy individual and social life in a community.” Christopher Alexander, The Oregon Experiment, Chapter 4
- “The purpose is... to help society and to help the community, and to contribute back to the community that we’re fortunate enough to do business in.” Jim Press, COO of Toyota Motor Sales in North America, Toyota Way, p. 72
- “Since Toyota’s founding we have adhered to the core principle of contributing to society through the practice of manufacturing high quality products and services” — Fujio Cho, in Liker, The Toyota Way, p. 35

Comparing Values

Serving the Community

Invest in the Community

Building on Experience

Slow decisions

根回し

Building on Experience

Feeling

Cash flow economics

Just-in-time

Reflection

Continuous Repair, Piecemeal Growth

Continuous Improvement

Geometry

Involving the community

- “The principle of participation: All decisions about what to build, and how to build it, will be in the hands of the users.” Christopher Alexander, The Oregon Experiment, Chapter 2
- “Nemawashi is the process of discussing proglems and potential solutions with all of those affected, to collect their ideas and get agreement on a path forward” — Liker, The Toyota Way, p. 40

Com

The principle of participation: All decisions about what to build, and how to build it, will be in the hands of the users. — Alexander

Nemawashi is the process of discussing problems and potential solutions with all of those affected, to collect their ideas and get agreement — Liker

Involving the Community

Involving the Community

Building on Experience

Slow decisions

根回し

Building on Experience

Feeling

Cash flow economics

Just-in-time

Reflection

Continuous Repair, Piecemeal Growth

Continuous Improvement

Geometry

Involving the community
- “The principle of participation: All decisions about what to build, and how to build it, will be in the hands of the users.” Christopher Alexander, The Oregon Experiment, Chapter 2
- “Nemawashi is the process of discussing proglems and potential solutions with all of those affected, to collect their ideas and get agreement on a path forward” — Liker, The Toyota Way, p. 40

Comparing Values

Involving the Community

Involving the Community

Building on Experience

Slow decisions

根回し

Building on Experience

Feeling

Cash flow economics

Just-in-time

Reflection

Continuous Repair, Piecemeal Growth

Continuous Improvement

Geometry

Involving the community
- “The principle of participation: All decisions about what to build, and how to build it, will be in the hands of the users.” Christopher Alexander, The Oregon Experiment, Chapter 2
- “Nemawashi is the process of discussing proglems and potential solutions with all of those affected, to collect their ideas and get agreement on a path forward” — Liker, The Toyota Way, p. 40

Comparing Values

Community

Building on Experience

Slow decisions

根回し

Building on Experience

Feeling

Cash flow economics

Just-in-time

Reflection

Continuous Repair, Piecemeal Growth

Continuous Improvement

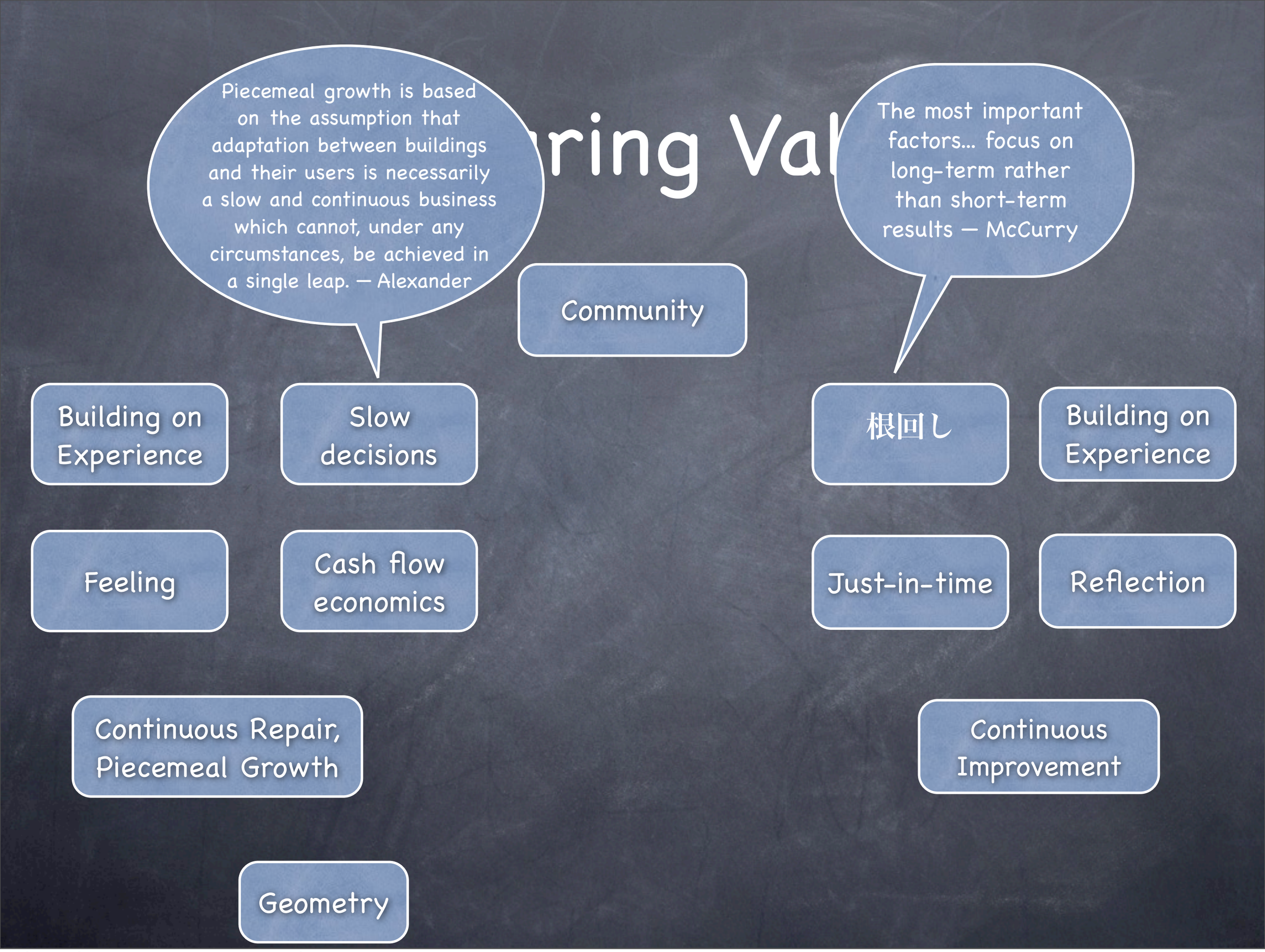
Geometry

Slow decisions / Nemawashi

“Piecemeal growth is based on the assumption that adaptation between buildings and their users is necessarily a slow and continuous business which cannot, under any circumstances, be achieved in a single leap.” – Christopher Alexander, The Oregon Experiment, p. 69

Nemawashi (Liker, The Toyota Way, p. 241)

“The most important factors for success are patience, a focus on long-term rather than short-term results, reinvestment in people, product, and plant, and an unforgiving commitment to quality.” — Robert B. McCurry, former Executive Vice President, Toyota Motor Sales. From Liker, The Toyota Way, p. 71.



Slow decisions / Nemawashi

“Piecemeal growth is based on the assumption that adaptation between buildings and their users is necessarily a slow and continuous business which cannot, under any circumstances, be achieved in a single leap.” – Christopher Alexander, The Oregon Experiment, p. 69

Nemawashi (Liker, The Toyota Way, p. 241)

“The most important factors for success are patience, a focus on long-term rather than short-term results, reinvestment in people, product, and plant, and an unforgiving commitment to quality.” — Robert B. McCurry, former Executive Vice President, Toyota Motor Sales. From Liker, The Toyota Way, p. 71.

Comparing Values

Community

Building on Experience

Slow decisions

根回し

Building on Experience

Feeling

Cash flow economics

Just-in-time

Reflection

Continuous Repair, Piecemeal Growth

Continuous Improvement

Geometry

Slow decisions / Nemawashi

“Piecemeal growth is based on the assumption that adaptation between buildings and their users is necessarily a slow and continuous business which cannot, under any circumstances, be achieved in a single leap.” – Christopher Alexander, The Oregon Experiment, p. 69

Nemawashi (Liker, The Toyota Way, p. 241)

“The most important factors for success are patience, a focus on long-term rather than short-term results, reinvestment in people, product, and plant, and an unforgiving commitment to quality.” — Robert B. McCurry, former Executive Vice President, Toyota Motor Sales. From Liker, The Toyota Way, p. 71.

Comparing Values

Community

Building on Experience

Deliberation

Building on Experience

Feeling

Cash flow economics

Just-in-time

Reflection

Continuous Repair, Piecemeal Growth

Continuous Improvement

Geometry

Building on Experience / Building on Experience
- “There is one timeless way of building. It is thousands of years old, and the same today as it has always been.” Christopher Alexander, The Timeless Way of Building, Ch. 1.
- Toyota Way Principle 8: Use only reliable, thoroughly tested technology that serves your people and processes. Liker, The Toyota Way, p. 166.

aring Value

There is one timeless way of building. It is thousands of years old, and the same today as it has always been. — Alexander

Toyota Way Principle 8: Use only reliable, thoroughly tested technology that serves your people and process

Building on Experience

Community

Deliberation

Building on Experience

Feeling

Cash flow economics

Just-in-time

Reflection

Continuous Repair, Piecemeal Growth

Continuous Improvement

Geometry

Building on Experience / Building on Experience
- “There is one timeless way of building. It is thousands of years old, and the same today as it has always been.” Christopher Alexander, The Timeless Way of Building, Ch. 1.
- Toyota Way Principle 8: Use only reliable, thoroughly tested technology that serves your people and processes. Liker, The Toyota Way, p. 166.

Comparing Values

Community

Building on Experience

Building on Experience

Deliberation

Feeling

Cash flow economics

Just-in-time

Reflection

Continuous Repair, Piecemeal Growth

Continuous Improvement

Geometry

Building on Experience / Building on Experience
- “There is one timeless way of building. It is thousands of years old, and the same today as it has always been.” Christopher Alexander, The Timeless Way of Building, Ch. 1.
- Toyota Way Principle 8: Use only reliable, thoroughly tested technology that serves your people and processes. Liker, The Toyota Way, p. 166.

Comparing Values

Community

Experience

Deliberation

Feeling

Cash flow economics

Just-in-time

Reflection

Continuous Repair, Piecemeal Growth

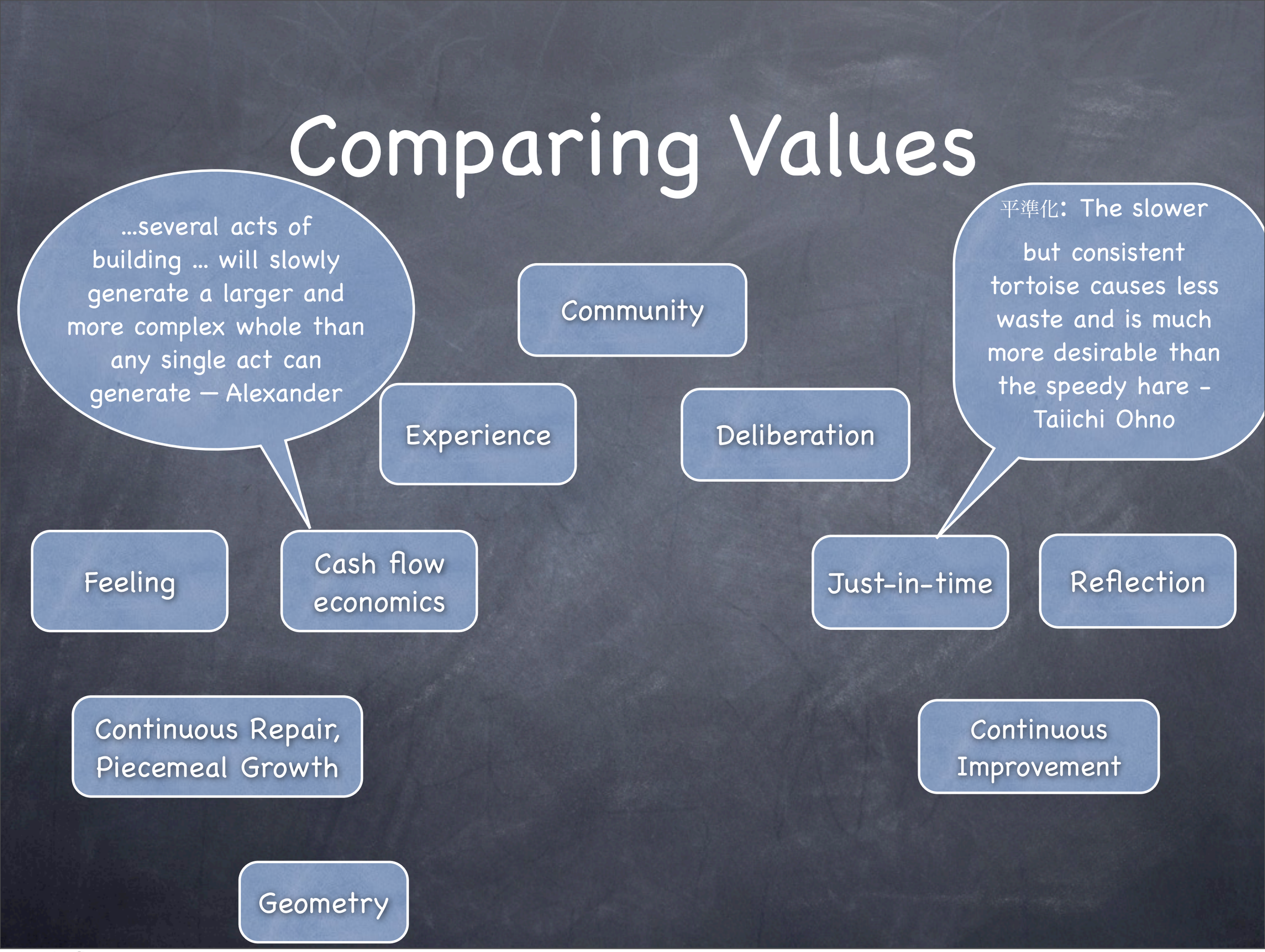
Continuous Improvement

Geometry

Cash flow economics / Just-in-time

- “Building takes place in increments, day by day, year by year. The people who live in the house... spend only what they can afford” - Grabow Christopher Alexander: The search for a new paradigm in architecture, p. 146
- “several acts of building, each one done to repair and magnify the product of the previous acts, will slowly generate a larger and more complex whole than any single act can generate.” - Christopher Alexander, The Timeless Way of Building, Ch. 24.

Hejunka (平準化) “The slower by consistent tortoise causes less waste and is much more desirable than the speedy hare that races ahead and then stops occasionally to doze. The Toyota Production System can be realized only when all the workers become tortoise (Ohno, 1988)” Taiichi Ohno, In Liker, The Toyota Way, p. 115



Cash flow economics / Just-in-time
- “Building takes place in increments, day by day, year by year. The people who live in the house... spend only what they can afford” - Grabow Christopher Alexander: The search for a new paradigm in architecture, p. 146
- “several acts of building, each one done to repair and magnify the product of the previous acts, will slowly generate a larger and more complex whole than any single act can generate.” - Christopher Alexander, The Timeless Way of Building, Ch. 24.

Hejunka (平準化) “The slower by consistent tortoise causes less waste and is much more desirable than the speedy hare that races ahead and then stops occasionally to doze. The Toyota Production System can be realized only when all the workers become tortoise (Ohno, 1988)” Taiichi Ohno, In Liker, The Toyota Way, p. 115

Comparing Values

Community

Experience

Deliberation

Feeling

Cash flow economics

Just-in-time

Reflection

Continuous Repair, Piecemeal Growth

Continuous Improvement

Geometry

Cash flow economics / Just-in-time

- “Building takes place in increments, day by day, year by year. The people who live in the house... spend only what they can afford” - Grabow Christopher Alexander: The search for a new paradigm in architecture, p. 146
- “several acts of building, each one done to repair and magnify the product of the previous acts, will slowly generate a larger and more complex whole than any single act can generate.” - Christopher Alexander, The Timeless Way of Building, Ch. 24.

Hejunka (平準化) “The slower by consistent tortoise causes less waste and is much more desirable than the speedy hare that races ahead and then stops occasionally to doze. The Toyota Production System can be realized only when all the workers become tortoise (Ohno, 1988)” Taiichi Ohno, In Liker, The Toyota Way, p. 115

Comparing Values

Community

Experience

Deliberation

Feeling

Incremental Development

Reflection

Continuous Repair, Piecemeal Growth

Continuous Improvement

Geometry

Feeling / Reflection

- And finally, of course, I want to paint a picture which allows me to understand the patterns of events which keep on happening in the thing whose structure I seek. In other words, I hope to find a picture, or a structure, which will, in some rather obvious and simple sense, account for the outward properties, for the pattern of events of the thing which I am studying." — The Timeless Way of Building, Chapter 5, 1979

“In Japan, sometimes the mother and the father say to the children, ‘Please do the Hansei.’ Some child did a bad thing. It means he or she must be sorry and improve his or her attitude—everything is included, spirit and attitude. So once the child is told, ‘Please do the Hansei,’ he understands almost everything about what the mother and the father want him to do.” — George Yamashina, Director of Toyota Technical Center. In Liker, The Toyota Way, p. 257.

Comparing Valu

What we need is a way of understanding the forces which cuts through this intellectual difficulty and goes closer to the empirical core.To do this., we must rely on feelings more than intellect. — Alexander

Hansei... means he or she must be sorry and improve his or her attitude—everything is included, spirit and attitude. So once the child is told, "Please do the Hansei," he understands almost everything about what the mother and the father want him to do. — George Yamashima

Community

Experience

Deliberation

Feeling

Incremental Development

Reflection

Continuous Repair, Piecemeal Growth

Continuous Improvement

Geometry

Feeling / Reflection

- And finally, of course, I want to paint a picture which allows me to understand the patterns of events which keep on happening in the thing whose structure I seek. In other words, I hope to find a picture, or a structure, which will, in some rather obvious and simple sense, account for the outward properties, for the pattern of events of the thing which I am studying." — The Timeless Way of Building, Chapter 5, 1979

“In Japan, sometimes the mother and the father say to the children, ‘Please do the Hansei.’ Some child did a bad thing. It means he or she must be sorry and improve his or her attitude—everything is included, spirit and attitude. So once the child is told, ‘Please do the Hansei,’ he understands almost everything about what the mother and the father want him to do.” — George Yamashina, Director of Toyota Technical Center. In Liker, The Toyota Way, p. 257.

Comparing Values

Feeling / Reflection

- And finally, of course, I want to paint a picture which allows me to understand the patterns of events which keep on happening in the thing whose structure I seek. In other words, I hope to find a picture, or a structure, which will, in some rather obvious and simple sense, account for the outward properties, for the pattern of events of the thing which I am studying." — The Timeless Way of Building, Chapter 5, 1979

“In Japan, sometimes the mother and the father say to the children, ‘Please do the Hansei.’ Some child did a bad thing. It means he or she must be sorry and improve his or her attitude—everything is included, spirit and attitude. So once the child is told, ‘Please do the Hansei,’ he understands almost everything about what the mother and the father want him to do.” — George Yamashina, Director of Toyota Technical Center. In Liker, The Toyota Way, p. 257.

Comparing Values

Community

Experience

Deliberation

Soulful Reflection

Incremental Development

Continuous Repair, Piecemeal Growth

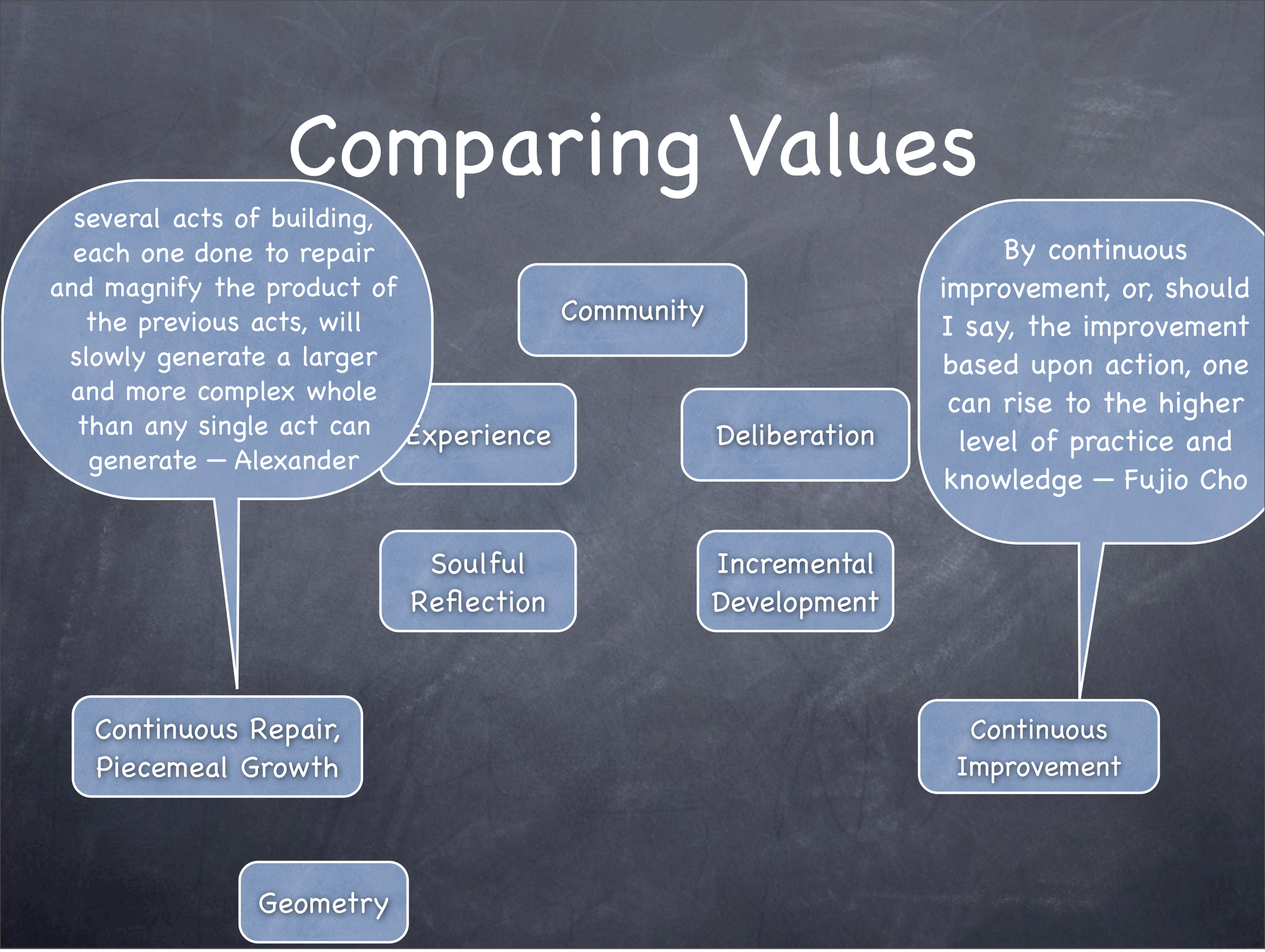
Continuous Improvement

Geometry

Continuous repair, piecemeal growth / Kaizen

“...several acts of building, each one done to repair and magnify the product of the previous acts, will slowly generate a larger and more complex whole than any single act can generate.” - Christopher Alexander, The Timeless Way of Building, Ch. 24.

“By continuous improvement, or, should I say, the improvement based upon action, one can rise to the higher level of practice and knowledge.” — Fujino Cho. In Liker, The Toyota Way, p. 3.



Continuous repair, piecemeal growth / Kaizen

“...several acts of building, each one done to repair and magnify the product of the previous acts, will slowly generate a larger and more complex whole than any single act can generate.” – Christopher Alexander, The Timeless Way of Building, Ch. 24.

“By continuous improvement, or, should I say, the improvement based upon action, one can rise to the higher level of practice and knowledge.” — Fujino Cho. In Liker, The Toyota Way, p. 3.

Comparing Values

Community

Experience

Deliberation

Soulful Reflection

Incremental Development

Continuous Repair, Piecemeal Growth

Continuous Improvement

Geometry

Continuous repair, piecemeal growth / Kaizen

“...several acts of building, each one done to repair and magnify the product of the previous acts, will slowly generate a larger and more complex whole than any single act can generate.” - Christopher Alexander, The Timeless Way of Building, Ch. 24.

“By continuous improvement, or, should I say, the improvement based upon action, one can rise to the higher level of practice and knowledge.” — Fujino Cho. In Liker, The Toyota Way, p. 3.

Community

Experience

Deliberation

Soulful Reflection

Incremental Development

Incremental Kaizen

Geometry

A leftover

Process

? Product

Geometry

A leftover...

Community

Experience

Deliberation

Soulful Reflection

Incremental Development

Incremental Kaizen

Geometry

A leftover

Process

? Product

Geometry

A leftover...

Process

Community

Experience

Deliberation

Soulful Reflection

Incremental Development

Incremental Kaizen

? Product

Geometry

A leftover

Process

? Product

Geometry

A leftover...

A leftover

Process

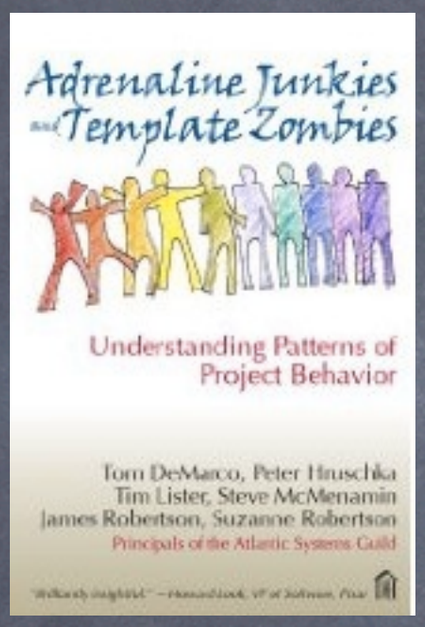
? Product

Geometry

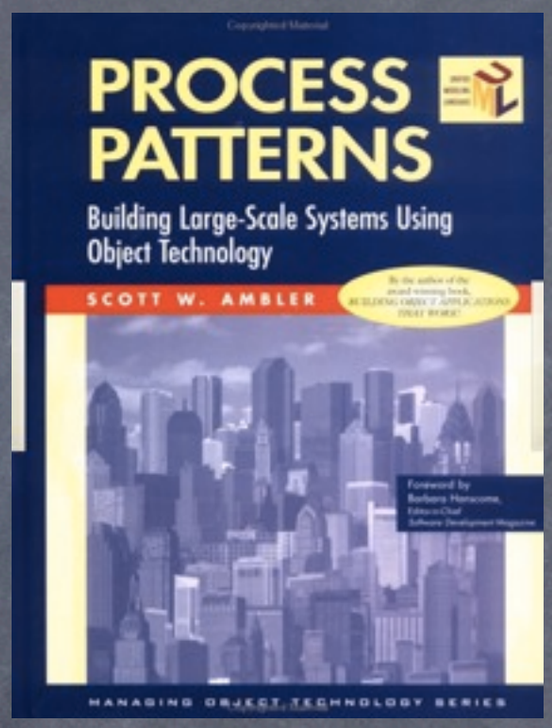
Today: Community?

Today: Community?
Tom DeMarco
Scott Ambler

Today: Community?



Tom DeMarco



Scott Ambler

Today: Community?
Tom DeMarco
Scott Ambler

How about the rest?

- Experienced pattern writers?
- Deliberation?
- Reflection with feeling?
- Measurable improvement from patterns?
- Any mature, proven practices in Agile?

➡ Software patterns have lost the vision

How about the rest?
Experienced pattern writers?
Deliberation?
Reflection with feeling?
Measurable improvement from patterns?
Any mature, proven practices in Agile?
Software patterns have lost the vision

2. Scrum and Agile

- Google Scrum + Agile = 2,000,000
- Google Scrum + Lean = 500,000
- But Scrum is Lean's child
- Is more or less ヨタ生産方式
- Great for production!

2. Scrum and Agile
Google Scrum + Agile = 2,000,000
Google Scrum + Lean = 500,000
But Scrum is Lean's child
Is more or less ヨタ生産方式 The Toyota Way
Great for production!

Scrum Foundations

- Lean is for complicated products
- Agile is for complex products
- Scrum is basically Lean, with Agile additions
- It is a good trick!

Scrum Foundations
Lean is for complicated projects
Agile is for complex products
Scrum is basically Lean, with Agile additions
It is a good trick!

Complex versus Complicated

- Agile: Dealing with complex systems: autopoeietic systems, self-organization; wholes greater than the sum of their parts
- Lean: Dealing with complicated systems. Building a car is complicated but not complex; the whole is the sum of its parts

Snowden and Boone, A Leader's Framework for Decision Making, Harvard Business Review, Nov. 2007

Complex versus complicated

Agile: Dealing with complex systems: autopoeietic systems, self-organization; wholes greater than the sum of their parts

Lean: Dealing with complicated systems. Building a car is complicated but not complex; the whole is the sum of its parts

Standards?

- Agile: Inspect and adapt: anyone can do it, you don't need to ask permission, you are empowered, and you achieve continuous improvement
- Lean: if you have a problem, spend up-front time seeking standards (Toyota Way, principle 6: Standardized Tasks are the Foundation for Continuous Improvement and Employee Empowerment)

Liker, Jeffrey K. The Toyota Way, McGraw-Hill, ©2004, Chapter 12, pp. 140 - 148

Standards

Agile: Inspect and adapt: anyone can do it, you don't need to ask permission, you are empowered, and you achieve continuous improvement

Lean: if you have a problem, spend up-front time seeking standards (Toyota Way, principle 6: Standardized Tasks are the Foundation for Continuous Improvement and Employee Empowerment)

Doing or Thinking?

- Agile: embrace change
- Lean: Long deliberation and thought with rapid deployment only after a decision is made (The Toyota Way, Principle 13: Make decisions slowly by consensus, thoroughly considering all options)

Liker, Jeffrey K. The Toyota Way, McGraw-Hill, ©2004, Chapter 19, pp. 237 - 249

Doing or Thinking?

Agile: embrace change

Lean: Long deliberation and thought with rapid deployment only after a decision is made (The Toyota Way, Principle 13: Make decisions slowly by consensus, thoroughly considering all options)

Specialization

- XP: No code ownership, no specialization. Scrum: cross-functional team
- Lean: spend years carefully grooming each individual to develop a depth of knowledge (from Toyota Way, Principle 10)

Liker, Jeffrey K. The Toyota Way, McGraw-Hill, ©2004, Chapter 16, pp. 184 - 198

Specialization

XP: No code ownership, no specialization. Scrum: cross-functional team

Lean: spend years carefully grooming each individual to develop a depth of knowledge (from Toyota Way, Principle 10)

Rework

- Agile: Refactoring compensates for architectural short-sightedness, maintenance, and emergent requirements (as well as keeping the code clean)
- Lean: Rework in design adds value, while rework in production is waste (Ballard: Negative Iteration, Lean Institute)

Ballard, Glenn, Positive vs Negative Iteration in Design. Lean construction Institute, University of California, Berkeley

Rework

Agile: Refactoring compensates for architectural short-sightedness, maintenance, and emergent requirements (as well as keeping the code clean)

Lean: Rework in design adds value, while rework in production is waste (Ballard: Negative Iteration, Lean Institute

Last Responsible Moment

- Agile: early decisions are likely to be wrong and cause rework, so defer to the last responsible moment
- Lean: letting a decision go beyond the point where it affects other decisions causes rework, so bring decisions forward to a point where their results don't propagate

Ballard, Glenn, Positive vs Negative Iteration in Design. Lean construction Institute, University of California, Berkeley

Last Responsible Moment

Agile: early decisions are likely to be wrong and cause rework, so defer to the last responsible moment

Lean: letting a decision go beyond the point where it affects other decisions causes rework, so bring decisions forward to a point where their results don't propagate

Patience versus Reaction

- "The most important factors for success are patience, a focus on long-term rather than short-term results, reinvestment in people, product, and plant, and an unforgiving commitment to quality."
— Robert B. McCurry, former Executive Vice President, Toyota Motor Sales

- A distinguished task force discovered that the number one reason for rocket launch failures was "responding to change over following a plan."
— Barry Boehm

Patience versus Reaction

"The most important factors for success are patience, a focus on long-term rather than short-term results, reinvestment in people, product, and plant, and an unforgiving commitment to quality." — Robert B. McCurry, former Executive Vice President, Toyota Motor Sales

A distinguished task force discovered that the number one reason for rocket launch failures was "responding to change over following a plan." — Barry Boehm

Missing from Scrum...

Missing from Scrum...
80% of software is maintenance
Toyota knows that: TPM
Hope for software:

Missing from Scrum...

- 80% of software is maintenance

Missing from Scrum...
80% of software is maintenance
Toyota knows that: TPM
Hope for software:

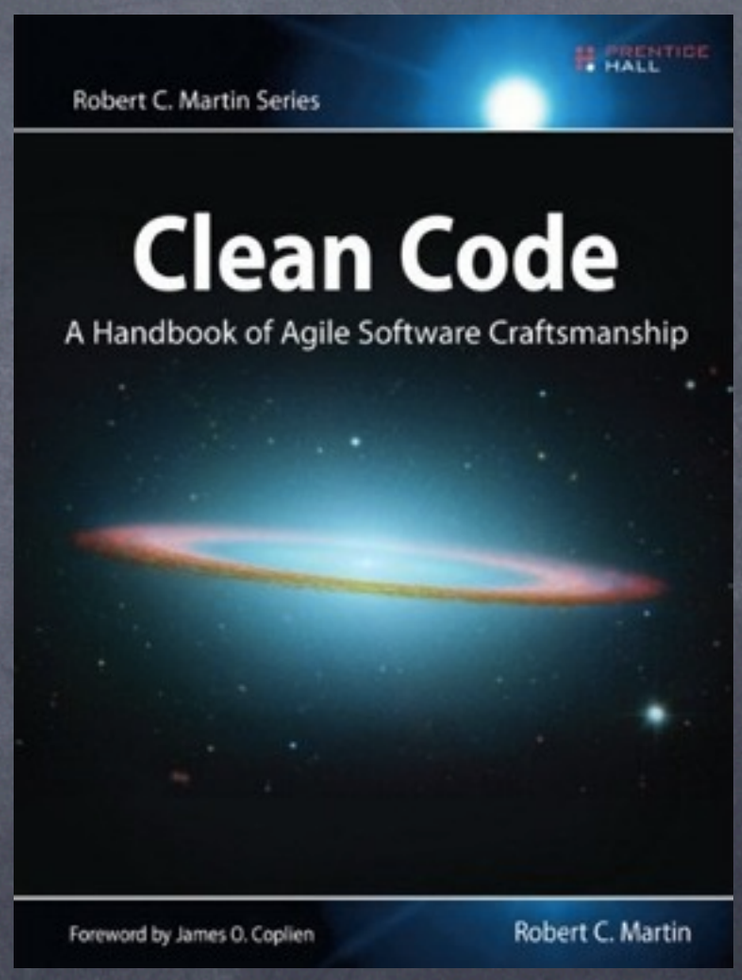
Missing from Scrum...

- 80% of software is maintenance
- Toyota knows that:
  総生産性メンテナンス

Missing from Scrum...
80% of software is maintenance
Toyota knows that: TPM
Hope for software:

Missing from Scrum...

- 80% of software is maintenance
- Toyota knows that:
  総生産性メンテナンス
- Hope in software:



Missing from Scrum...
80% of software is maintenance
Toyota knows that: TPM
Hope for software:

Summary of Scrum and Agile

- The Scrum & Lean bits
  - Scrum values planning
  - Strong on eliminating muda, mura, muri
  - Process and product
  - Cross-functional teams
  - Even deeper roots in Buddhism
- The Agile bits
  - Much doing, little deliberation
  - Kaizen in "red pill Scrum," but few do that
  - Tends to draw attention to features rather than architecture
  - Most teams use fads instead of proven practices

Summary of Scrum and Agile
The Lean bits
Value planning
Strong on eliminating muda, mura, muri
Process and product
Cross-functional teams
Even deeper roots in Buddhism
The Agile bits
Much doing, little deliberation
Kaizen in "red pill Scrum," but few do that
Most teams use fads instead of proven practices
Tends to draw attention to features rather than architecture

Siniaalto and Abrahamsson. Does Test-Driven Development Improve the Program Code? Alarming results from a Comparative Case Study. Proceedings of Cee-Set 2007, 10 - 12 October, 2007, Poznan, Poland.

Siniaalto and Abrahamsson, Comparative Case Study on the Effect of Test-Driven Development on Program Design and Test Coverage, ESEM 2007.

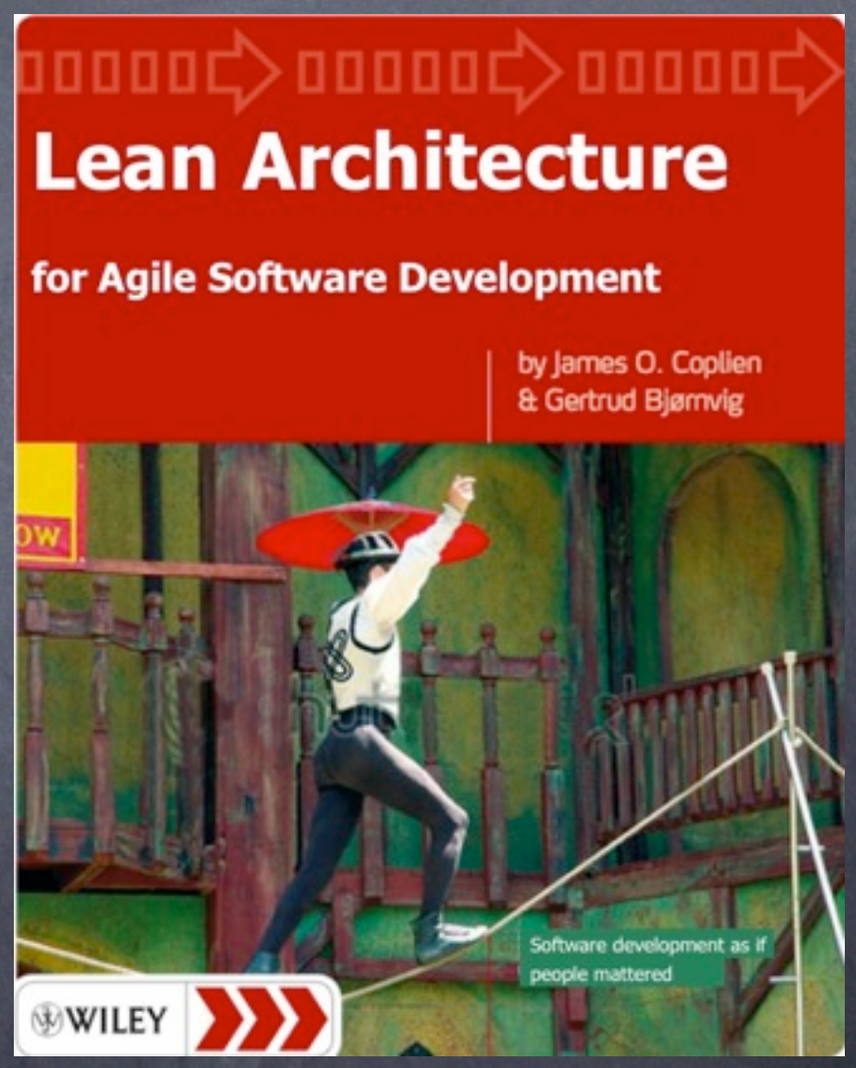
Bell & Woit, "Integration of Social-Psychological Influence Techniques into the XP Negotiation Process", unpublished research. Cited with permission (verbal permission obtained from Kathy Bell on 12 September 2006).

Martin, Angela, R. Biddle and J. Noble. The XP Customer Role in Practice: Three Case Studies. Proceedings of the Second Agile Development Conference, 2004.

Martin, Angela. Exploring the XP Customer Role - Part II. Proceedings of the Fifth International Conference on eXtreme Programming and Agile Processes in Software Engineering, Jutta Eckstein and Hubert Baumeister, eds., 2004.

3. Lean Architecture and DCI

- Process focus of Lean – but also product
- Main focus:
  - Planning and Thinking
  - Pull
  - Removing 斑
  - ポカヨケ
  - Reduced 無駄
  - Just-in-time
  - One-piece continuous flow

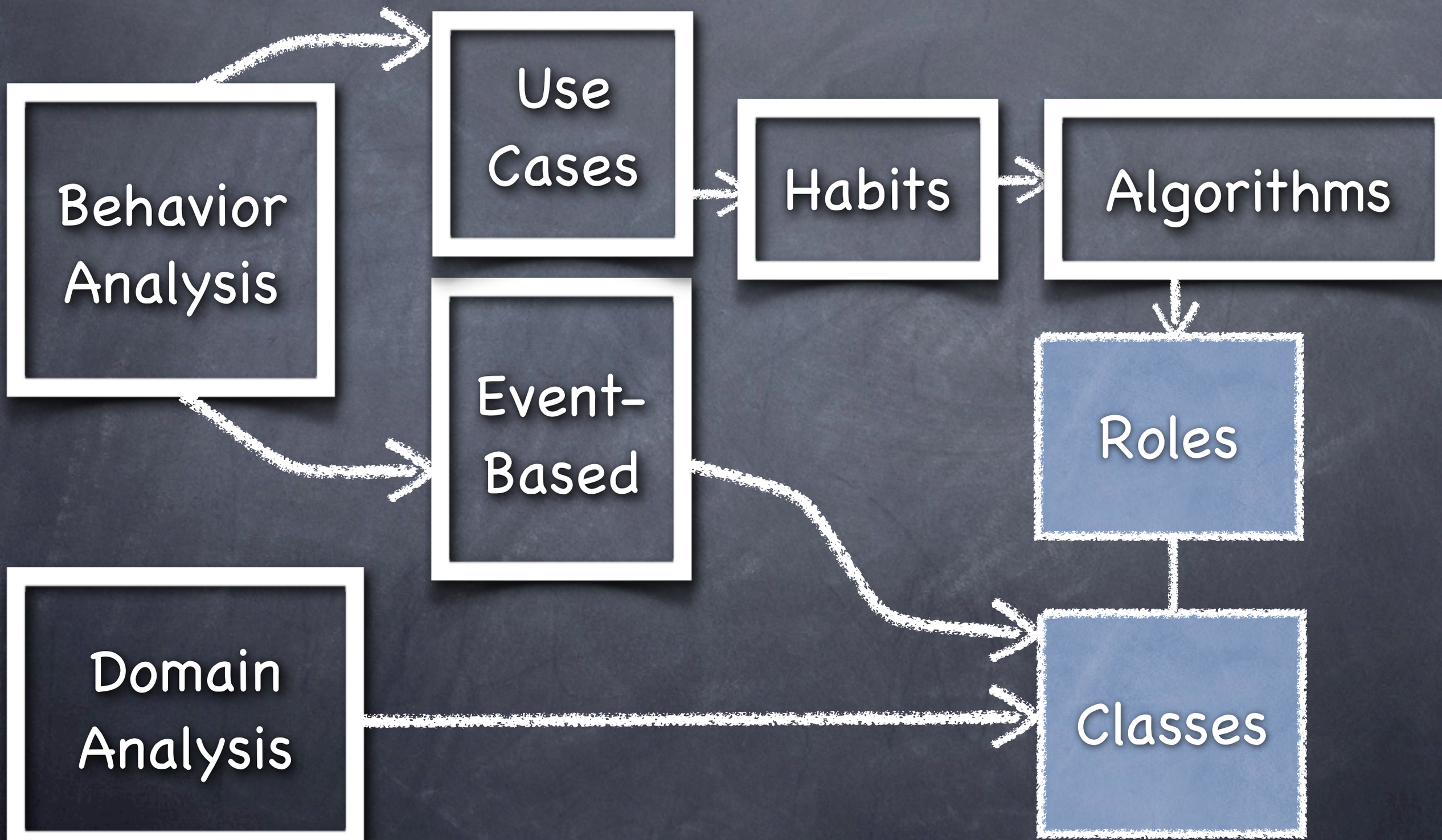


http://www.leansoftwarearchitecture.com

Process focus of Lean — but also product

Main focus:
- Planning and thinking
- Pull
- Removing inconsistency (mura)
- Poka-yoke
- Reduced waste (muda)
- Just-in-time
- One-piece continuous flow

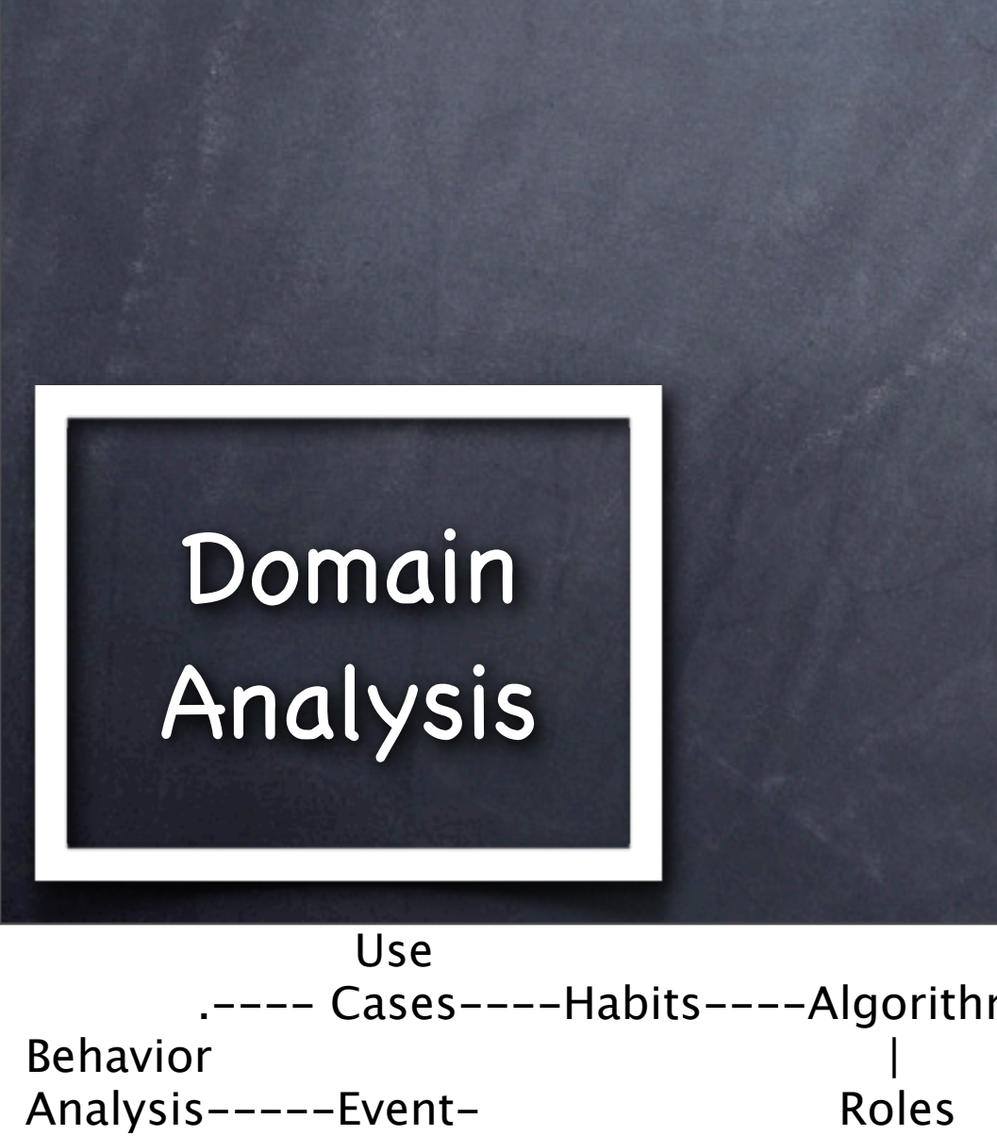
The development process




```
                Use
.---- Cases----Habits----Algorithms
Behavior                    |
Analysis-----Event-         Roles
             based          |
               |            |
Domain---------------------Classes
Analysis
```

The development process

Domain Analysis



Lean Architecture: Domain Engineering

- Experience
- Geometry
- Cross-functional team

Lean Architecture: Domain Engineering
Experience
Geometry
Cross-functional team

Two kinds of OO

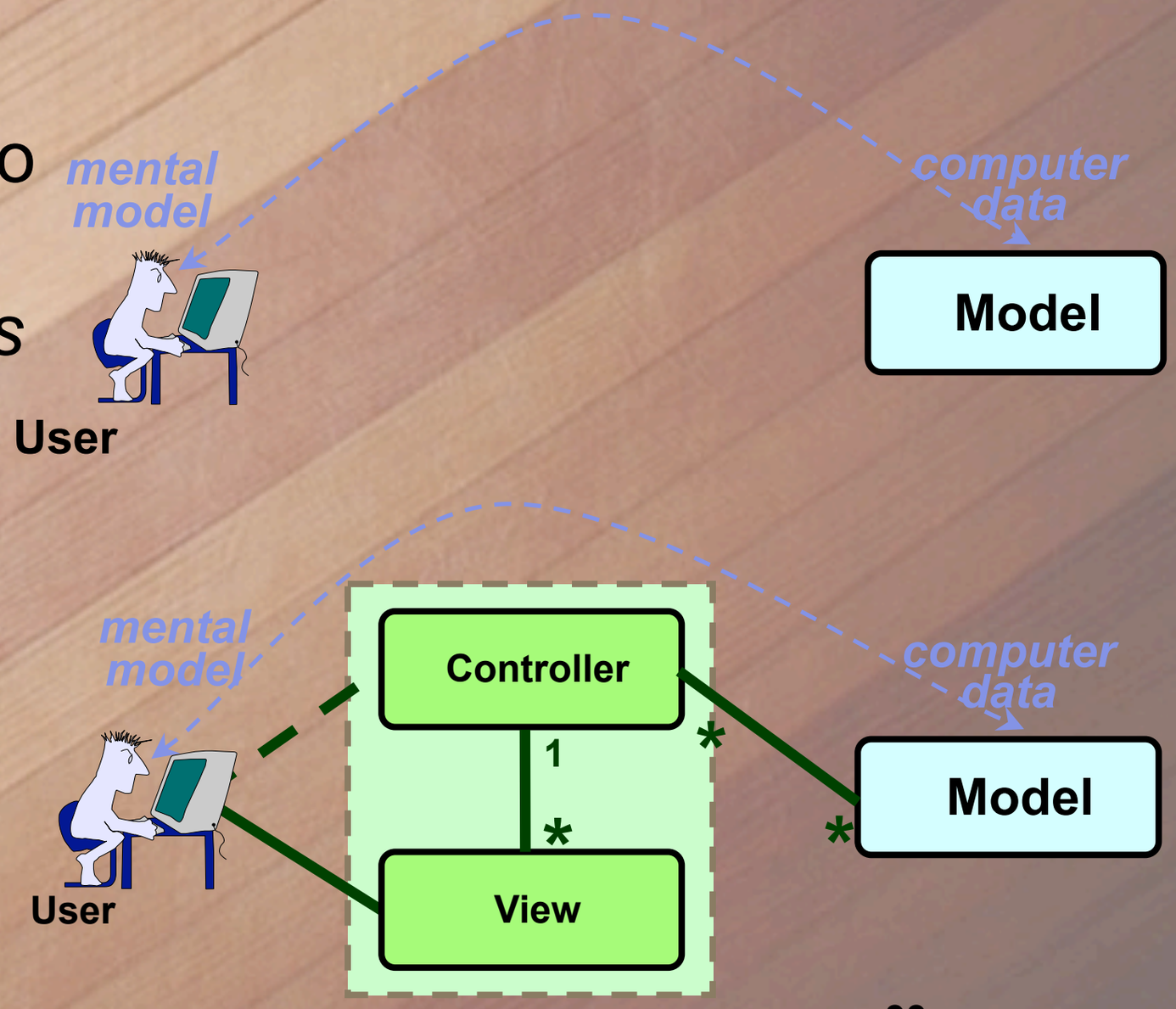
Concern	Atomic Event	DCI Architecture
User Goal	Direct manipulation of a domain object	A sequence of tasks toward a goal
Requirements	State machine, custom formalism	Use Case
Technology	Good old OO	Multiparadigm design DCI
Design focus	Form of the data	Form of the algorithm
Scope	Single primary object or small static network	Multiple objects with dynamic associations
Interaction	Noun-verb	Verb-noun
Example	Delete character Print balance	Spell check Money transfer

Concern	Atomic Event	DCI Architecture
User Goal	Direct domain object manipulation	A sequence of tasks toward a goal
Requirements	State machine, custom formalism	Use Case
Technology	Good old OO	Multi-paradigm design; DCI
Design focus	Form of the data	Form of the algorithm
Scope	Single primary object s or small static network of object	Multiple objects with dynamic associations
Interaction	Noun-verb	Verb-noun
Example	Delete character Print balance	Spell check Money transfer

MVC: The Embodiment of the OO Vision

- User model -> into the code -> presented back to the user
- The goal of *views* is direct manipulation

The goal of the *controller* is to coordinate multiple views



MVC: The Embodiment of the OO Vision
- User model -> into the code -> presented back to the user
- The goal of *views* is direct manipulation
- The goal of the *controller* is to coordinate multiple views

Model

Controller

User / mental model Model

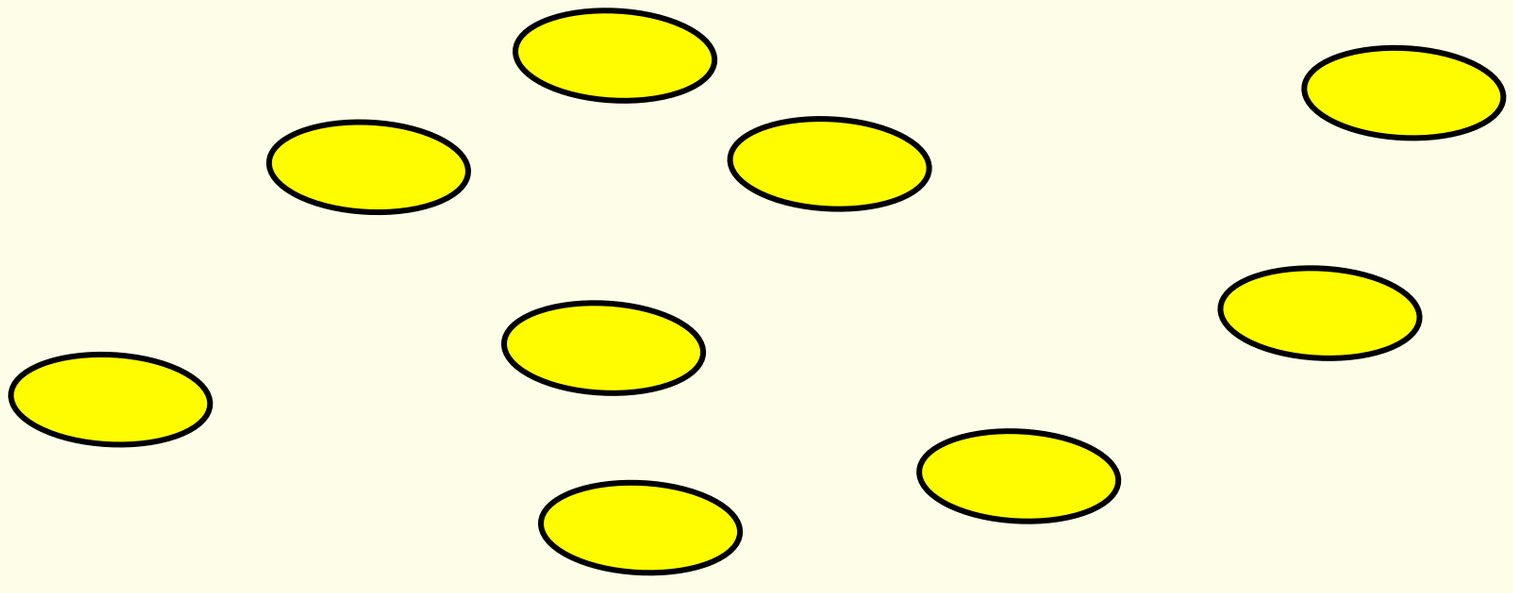
View

The end user mental model

- On the other hand, people need a chance to identify with the part of the environment in which they live and work; they want some sense of ownership, some sense of territory. The most vital question about the various places in any community is always this: Do the people who use them own them psychologically? Do they feel that they can do with them as they they wish; do they feel that the place is theirs; are they free to make the place their own? — Christopher Alexander

On the other hand, people need a chance to identify with the part of the environment in which they live and work; they want some sense of ownership, some sense of territory. The most vital question about the various places in any community is always this: Do the people who use them own them psychologically? Do they feel that they can do with them as they they wish; do they feel that the place is theirs; are they free to make the place their own? Christopher Alexander, The Oregon Experiment, p. 38

System Operations
Separation of Concern



Computers can:

store and retrieve data

+ transform data

+ ***communicate!!!***

Communication becomes first class citizen in computing

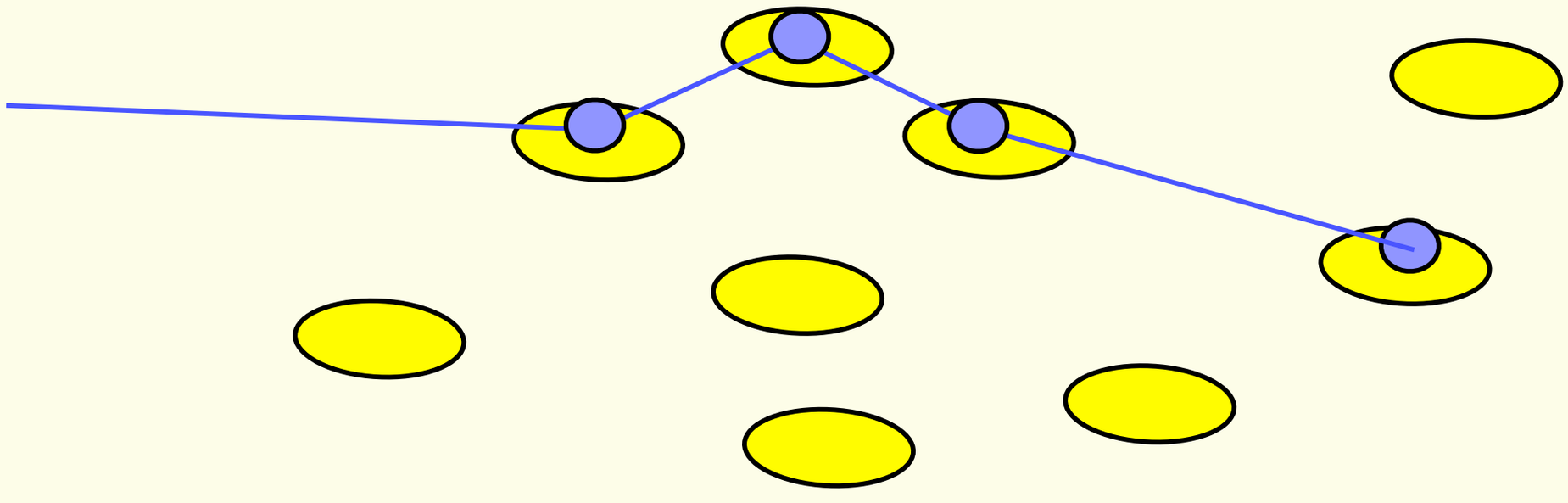
Show execution of the three tasks (on clicks)

Class oriented programming: NOODLES

DCI: Separation of concern:

Each task coded separately (animated on clicks)

System Operations
Separation of Concern



Computers can:

store and retrieve data

+ transform data

+ ***communicate!!!***

Communication becomes first class citizen in computing

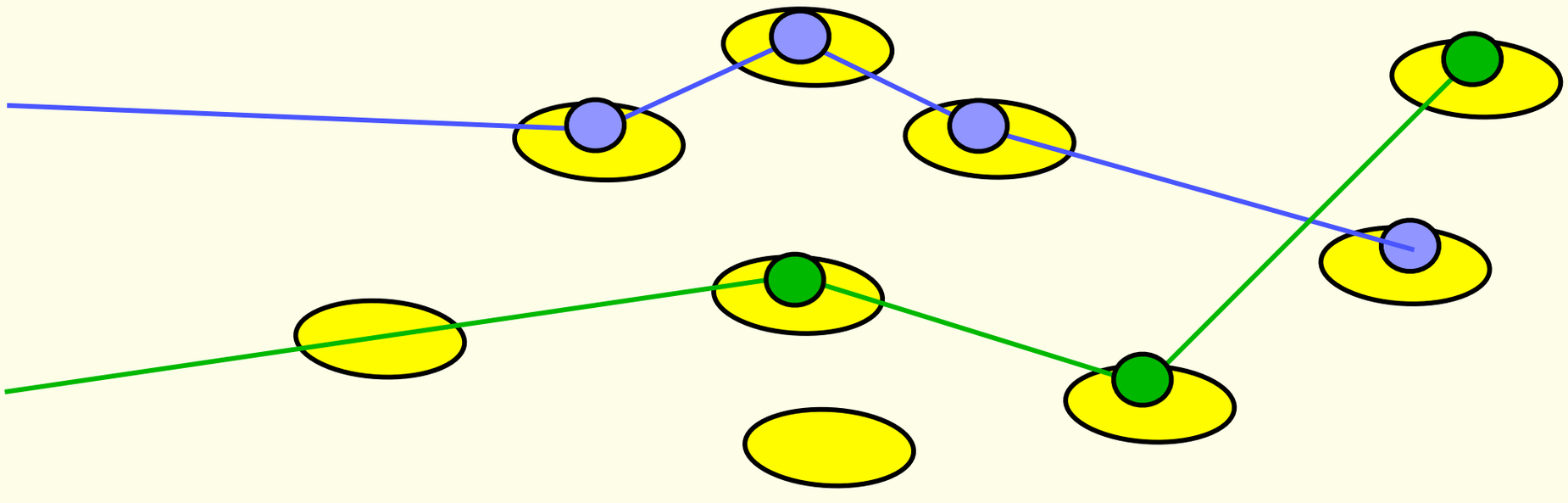
Show execution of the three tasks (on clicks)

Class oriented programming: NOODLES

DCI: Separation of concern:

Each task coded separately (animated on clicks)

System Operations
Separation of Concern



Computers can:

store and retrieve data

+ transform data

+ ***communicate!!!***

Communication becomes first class citizen in computing

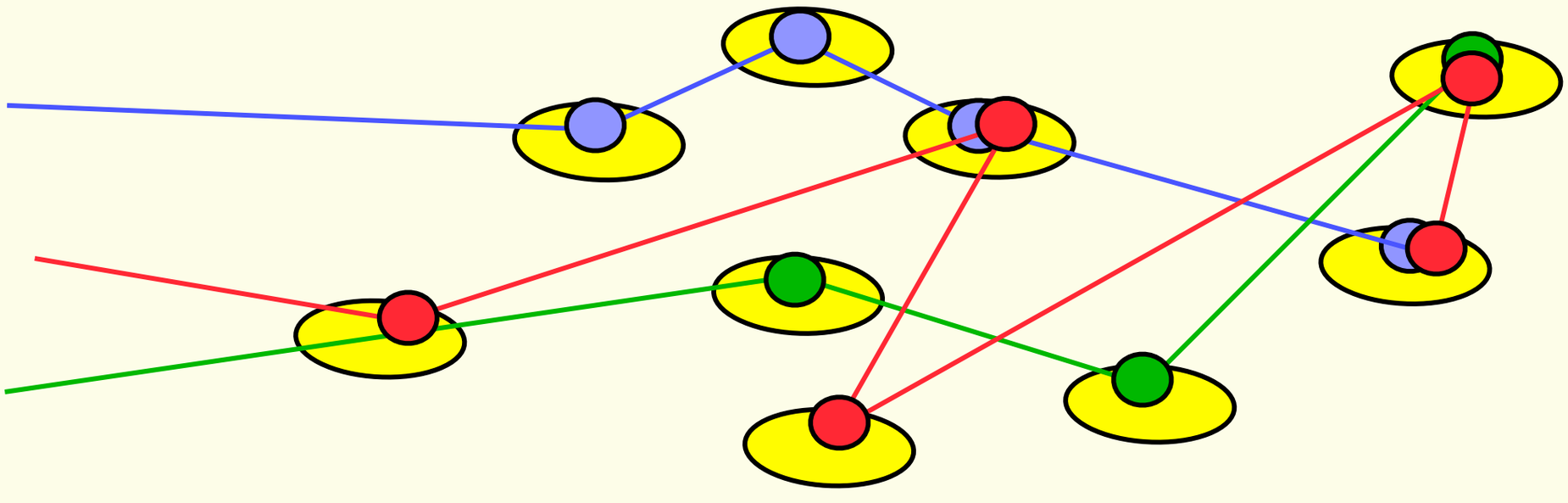
Show execution of the three tasks (on clicks)

Class oriented programming: NOODLES

DCI: Separation of concern:

Each task coded separately (animated on clicks)

System Operations
Separation of Concern



Computers can:

store and retrieve data

+ transform data

+ ***communicate!!!***

Communication becomes first class citizen in computing

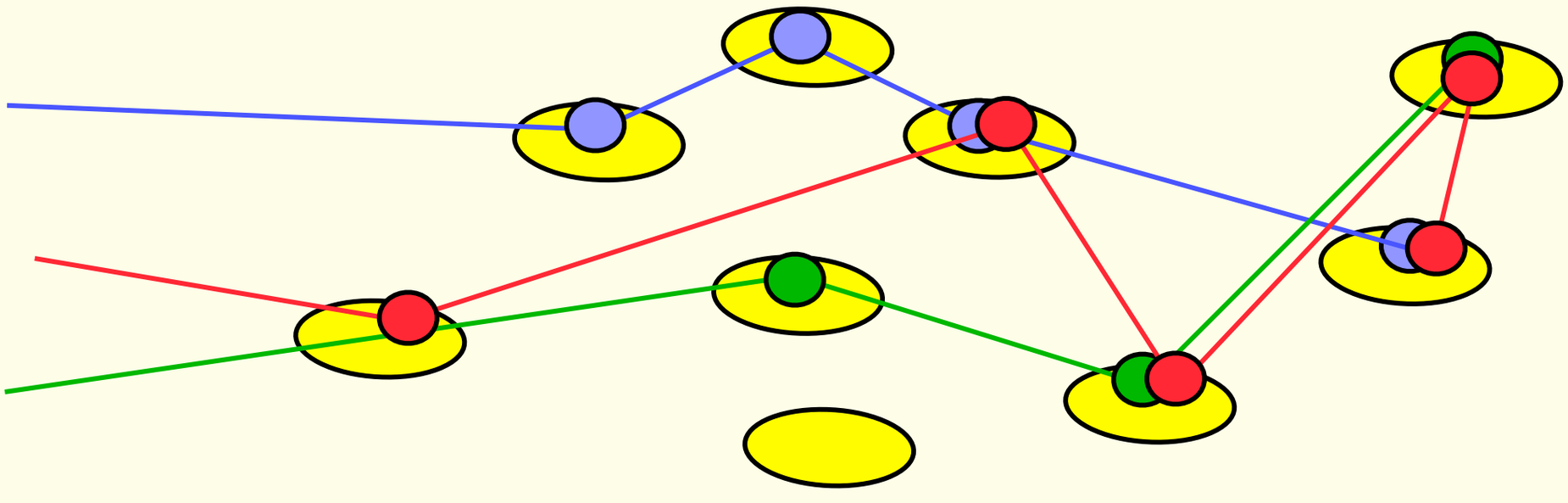
Show execution of the three tasks (on clicks)

Class oriented programming: NOODLES

DCI: Separation of concern:

Each task coded separately (animated on clicks)

System Operations
Separation of Concern



Computers can:

store and retrieve data

+ transform data

+ ***communicate!!!***

Communication becomes first class citizen in computing

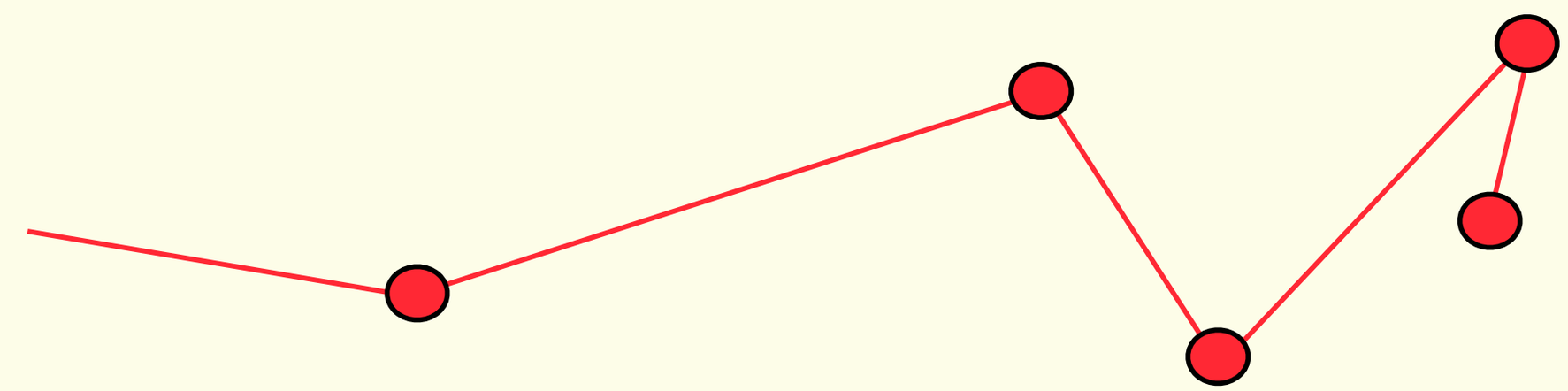
Show execution of the three tasks (on clicks)

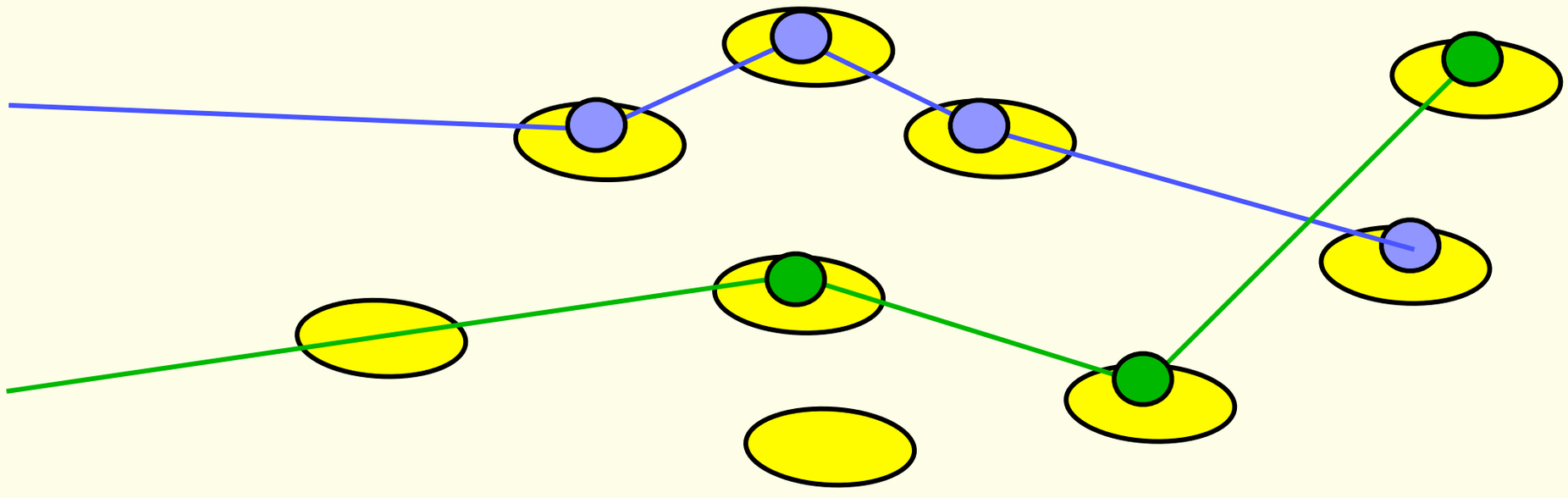
Class oriented programming: NOODLES

DCI: Separation of concern:

Each task coded separately (animated on clicks)

System Operations
Separation of Concern





Computers can:

store and retrieve data

+ transform data

+ ***communicate!!!***

Communication becomes first class citizen in computing

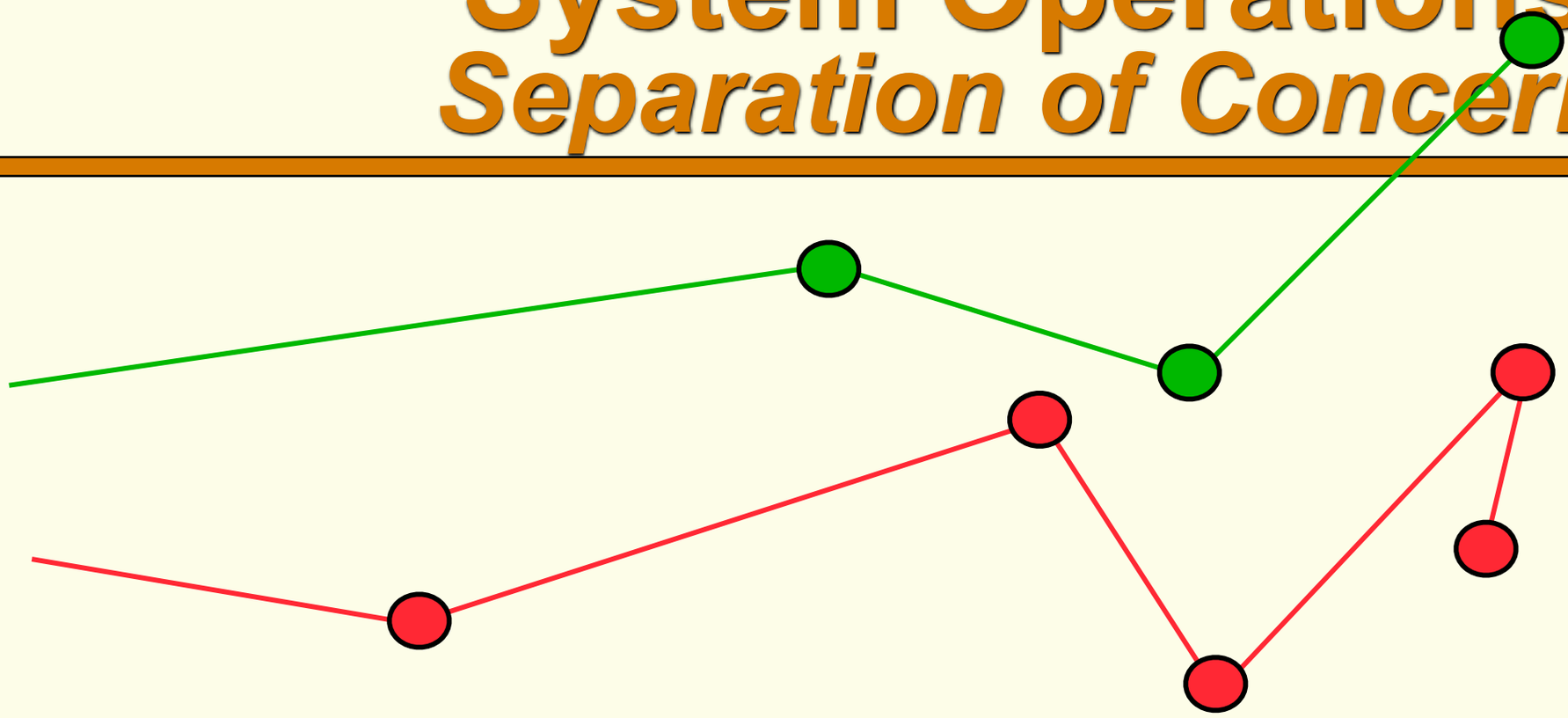
Show execution of the three tasks (on clicks)

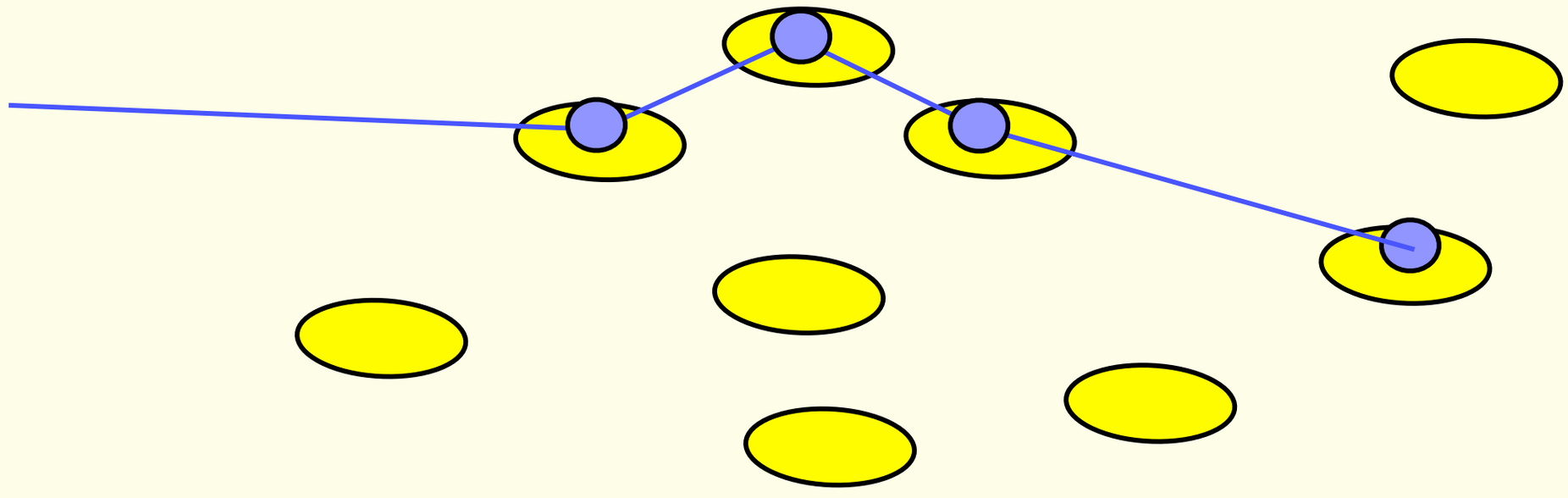
Class oriented programming: NOODLES

DCI: Separation of concern:

Each task coded separately (animated on clicks)

System Operations
Separation of Concern





Computers can:

store and retrieve data

+ transform data

+ ***communicate!!!***

Communication becomes first class citizen in computing

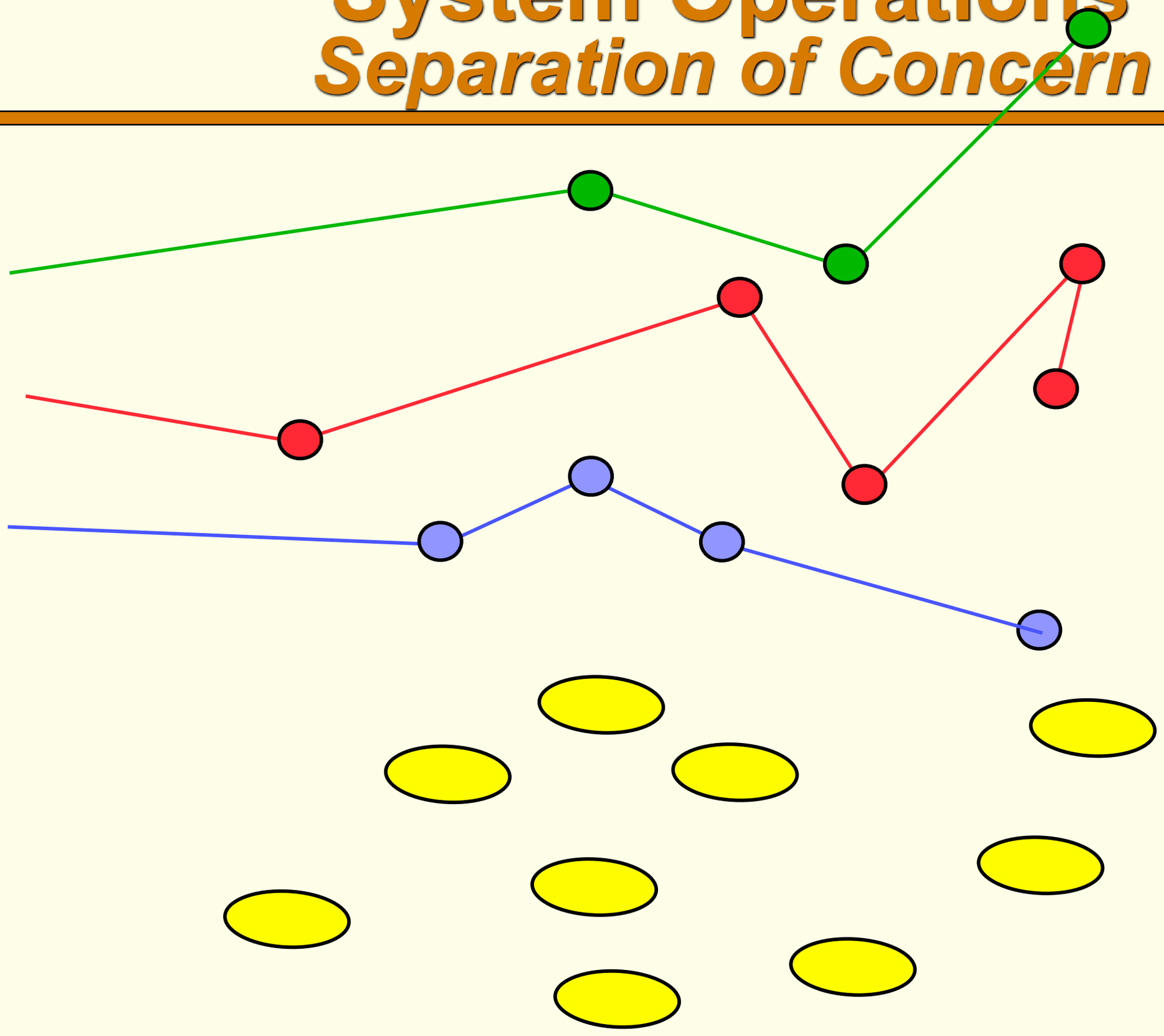
Show execution of the three tasks (on clicks)

Class oriented programming: NOODLES

DCI: Separation of concern:

Each task coded separately (animated on clicks)

System Operations
Separation of Concern



Computers can:

store and retrieve data

+ transform data

+ ***communicate!!!***

Communication becomes first class citizen in computing

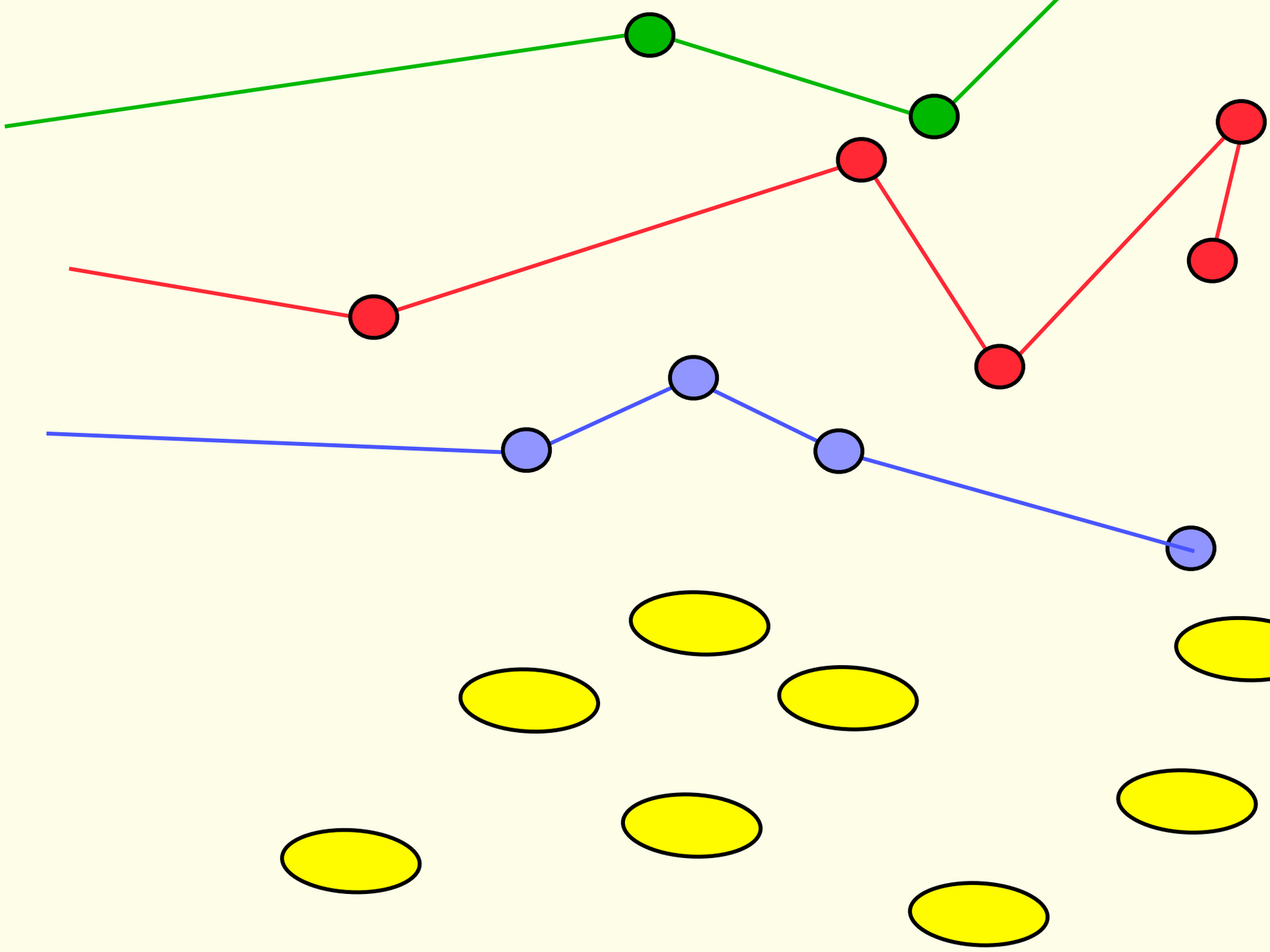
Show execution of the three tasks (on clicks)

Class oriented programming: NOODLES

DCI: Separation of concern:

Each task coded separately (animated on clicks)

System Operations
Executed by Contexts



A Context (an instance of a context class)
receives a message
is responsible for a use case/task/system operation
triggers a method in the first role
execution continues as specified in role methods

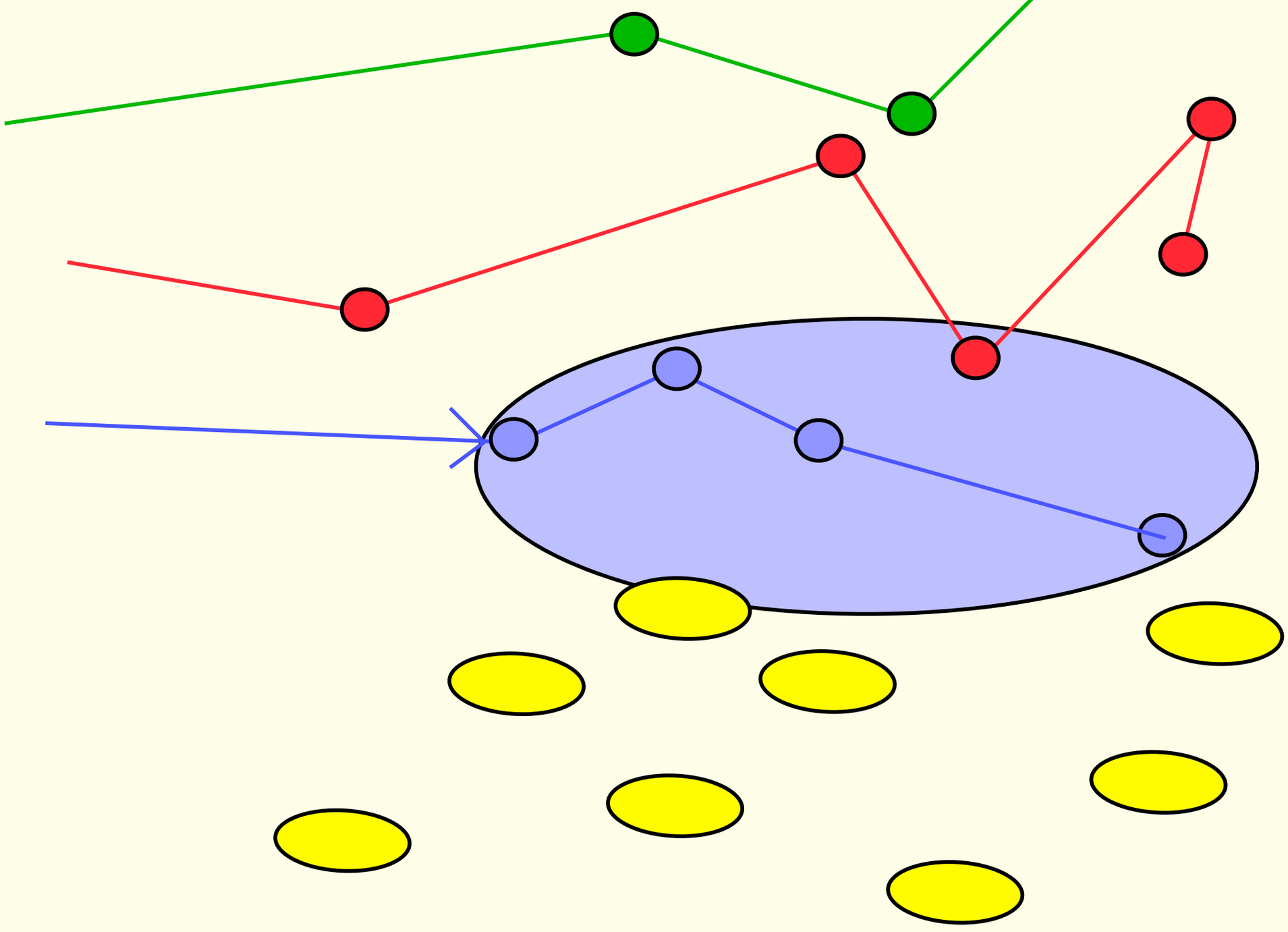
functional decomposition context responsibility/role responsibilities

50 years ago:
Data store / applications / functional decomposition
+ Red circles: access routines

DCI:
Data objects / contexts / methodful roles
+ Context binds roles to objects:
mobilize the objects that actually do the work

NOTE
Binding role/object not shown,
late (runtime) binding

System Operations
Executed by Contexts



A Context (an instance of a context class)
receives a message
is responsible for a use case/task/system operation
triggers a method in the first role
execution continues as specified in role methods

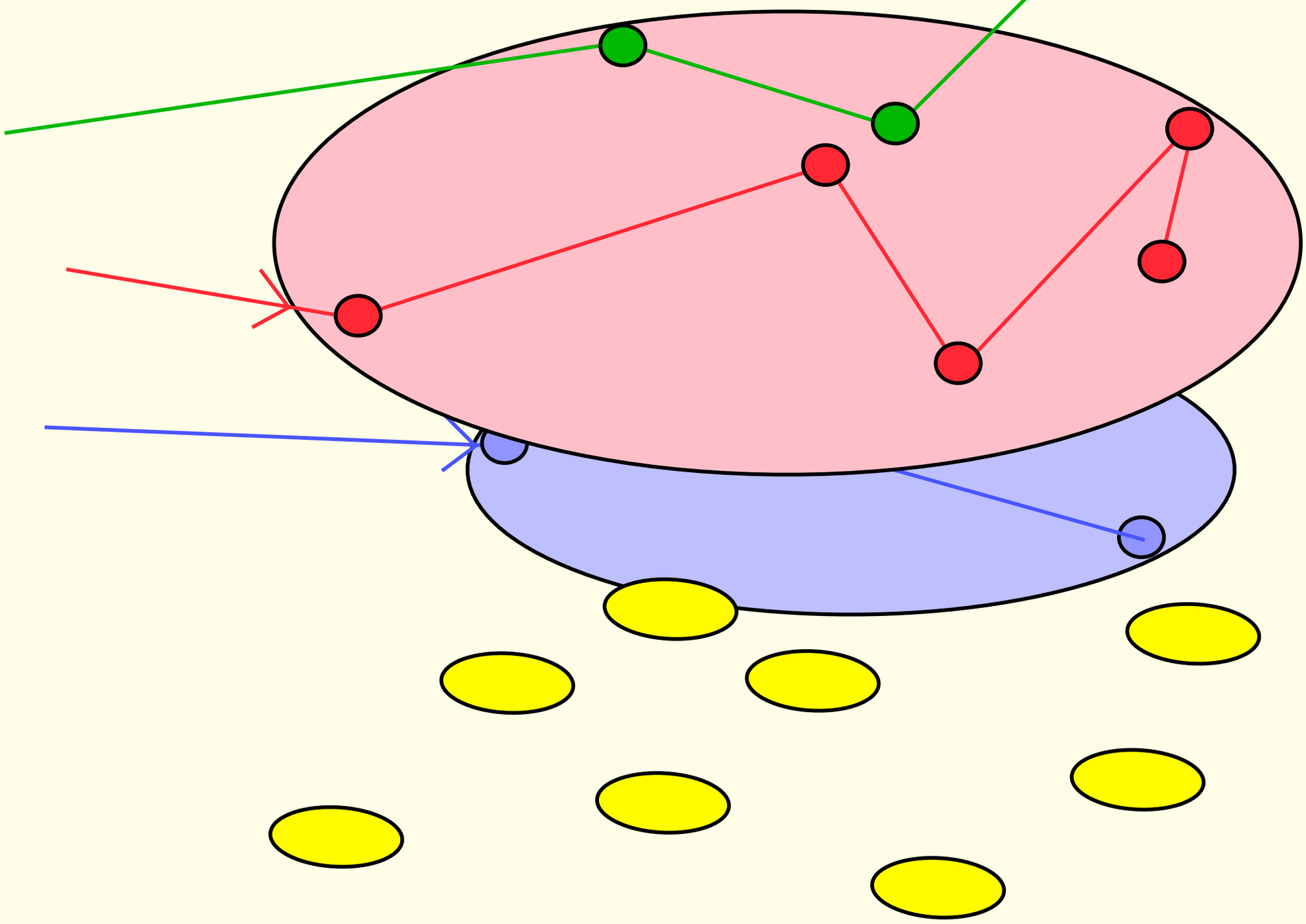
functional decomposition context responsibility/role responsibilities

50 years ago:
Data store / applications / functional decomposition
+ Red circles: access routines

DCI:
Data objects / contexts / methodful roles
+ Context binds roles to objects:
mobilize the objects that actually do the work

NOTE
Binding role/object not shown,
late (runtime) binding

System Operations
Executed by Contexts



A Context (an instance of a context class)
receives a message
is responsible for a use case/task/system operation
triggers a method in the first role
execution continues as specified in role methods

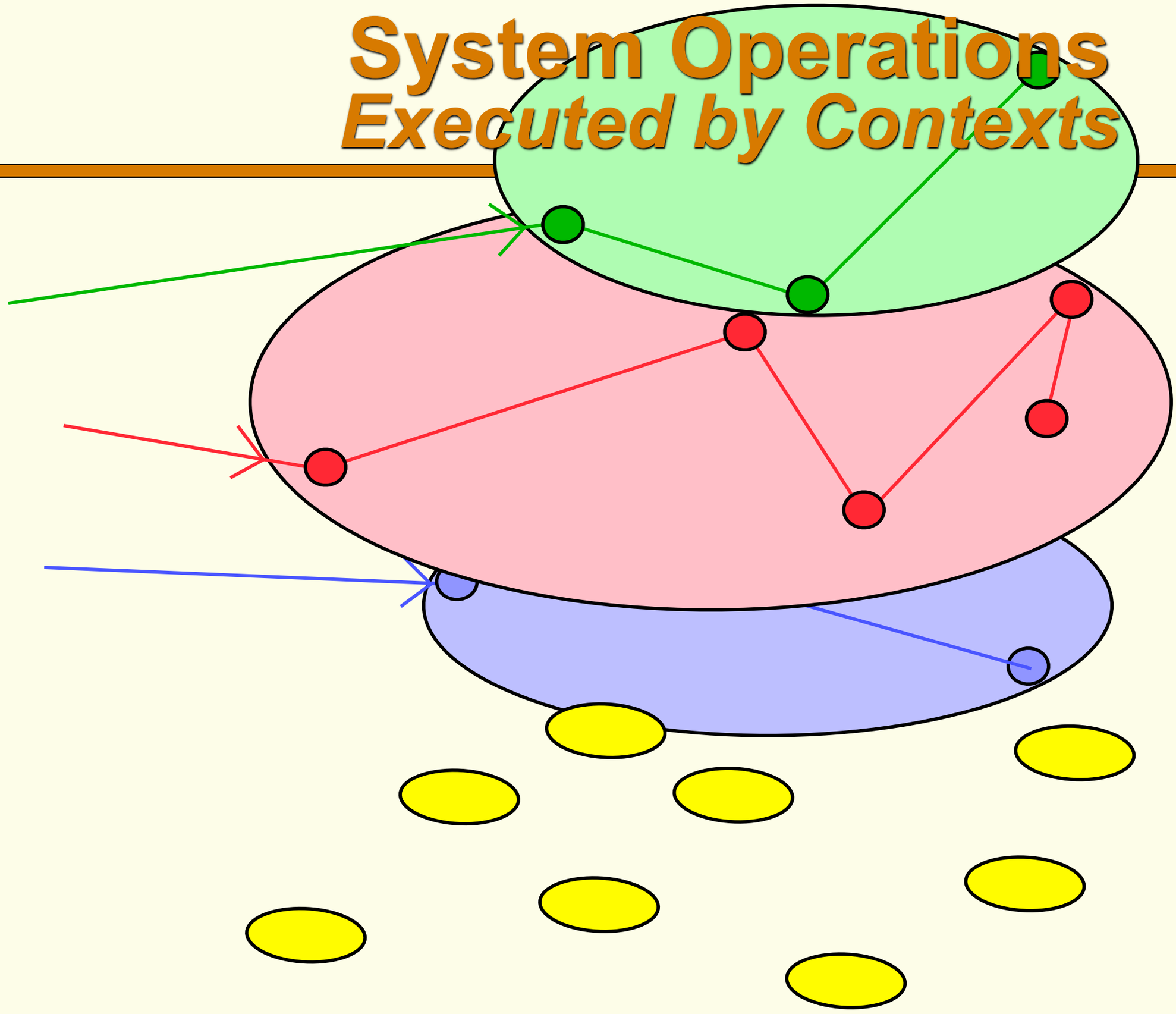
functional decomposition context responsibility/role responsibilities

50 years ago:
Data store / applications / functional decomposition
+ Red circles: access routines

DCI:
Data objects / contexts / methodful roles
+ Context binds roles to objects:
mobilize the objects that actually do the work

NOTE
Binding role/object not shown,
late (runtime) binding

System Operations
Executed by Contexts



A Context (an instance of a context class)
receives a message
is responsible for a use case/task/system operation
triggers a method in the first role
execution continues as specified in role methods

functional decomposition context responsibility/role responsibilities

50 years ago:
Data store / applications / functional decomposition
+ Red circles: access routines

DCI:
Data objects / contexts / methodful roles
+ Context binds roles to objects:
mobilize the objects that actually do the work

NOTE
Binding role/object not shown,
late (runtime) binding

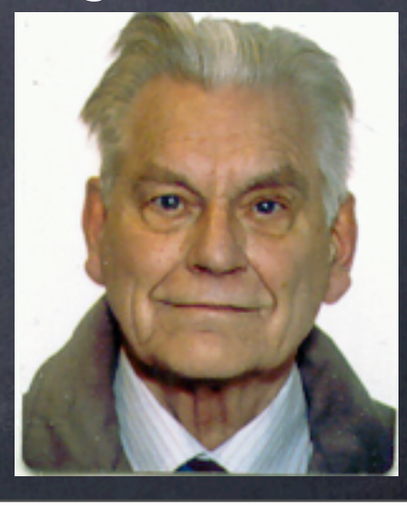
And finally, of course, I want to paint a picture which allows me to understand the patterns of events which keep on happening in the thing whose structure I seek. In other words, I hope to find a picture, or a structure, which will, in some rather obvious and simple sense, account for the outward properties, for the pattern of events of the thing which I am studying.

— Christopher Alexander, The Timeless Way of Building, Chapter 5, 1979

And finally, of course, I want to paint a picture which allows me to understand the patterns of events which keep on happening in the thing whose structure I seek. In other words, I hope to find a picture, or a structure, which will, in some rather obvious and simple sense, account for the outward properties, for the pattern of events of the thing which I am studying.
— Christopher Alexander, The Timeless Way of Building, Chapter 5, 1979

And finally, of course, I want to paint a picture which allows me to understand the patterns of events which keep on happening in the thing whose structure I seek. In other words, I hope to find a picture, or a structure, which will, in some rather obvious and simple sense, account for the outward properties, for the pattern of events of the thing which I am studying.

— Christopher Alexander, The Timeless Way of Building, Chapter 5, 1979



And finally, of course, I want to paint a picture which allows me to understand the patterns of events which keep on happening in the thing whose structure I seek. In other words, I hope to find a picture, or a structure, which will, in some rather obvious and simple sense, account for the outward properties, for the pattern of events of the thing which I am studying.
— Christopher Alexander, The Timeless Way of Building, Chapter 5, 1979

Some pattern ideas in DCI

- Always a human element: the mental model
- Local symmetry breaking:
  - Use cases into methods
  - Businesses into roles
- Systems by composition (e.g., roles and classes)
- General overall form, a million variants

Some pattern ideas in DCI:
Always a human element: the mental model
Local symmetry breaking:
Use cases into methods
Businesses into roles
Systems by composition (e.g., roles and classes)
General overall form, a million variants

DCI: Symmetry Breaking

Phone call

DCI: Symmetry breaking.
Phone Call

The HOPP Pattern

Half-call Half-call

Global symmetry

The HOPP Pattern
Half-call Half-call
Global Symmetry

DCI: Adding Roles

Calling Party
(Role)

Called Party
(Role)

Terminal

Terminal

Local symmetries

DCI: Adding Roles

Calling Party
(Role)

Called Party
(Role)

Terminal

Terminal

Local Symmetries

The human perspective

The human perspective
My phone, me, her, phone call

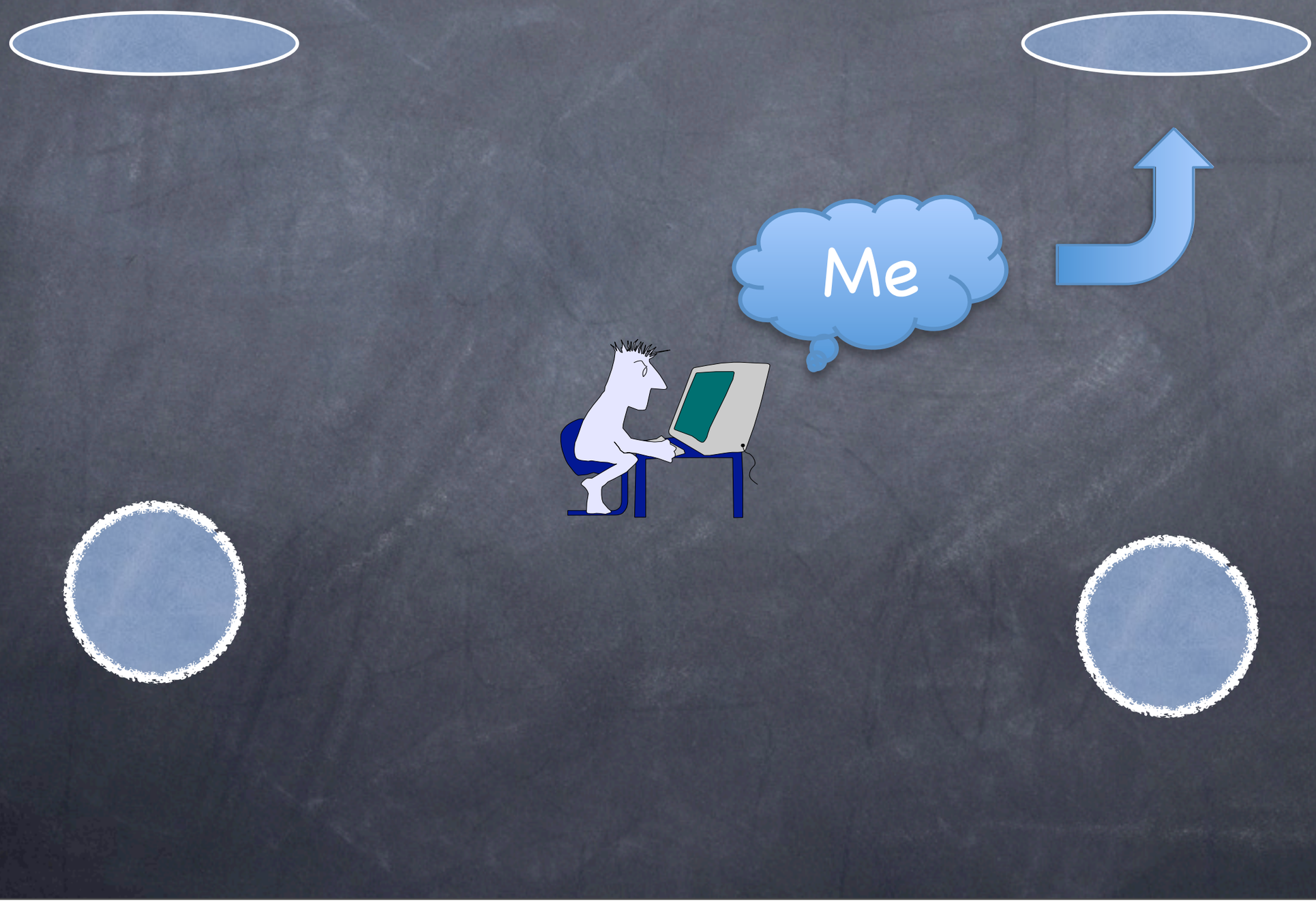
The human perspective

The human perspective
My phone, me, her, phone call

The human perspective

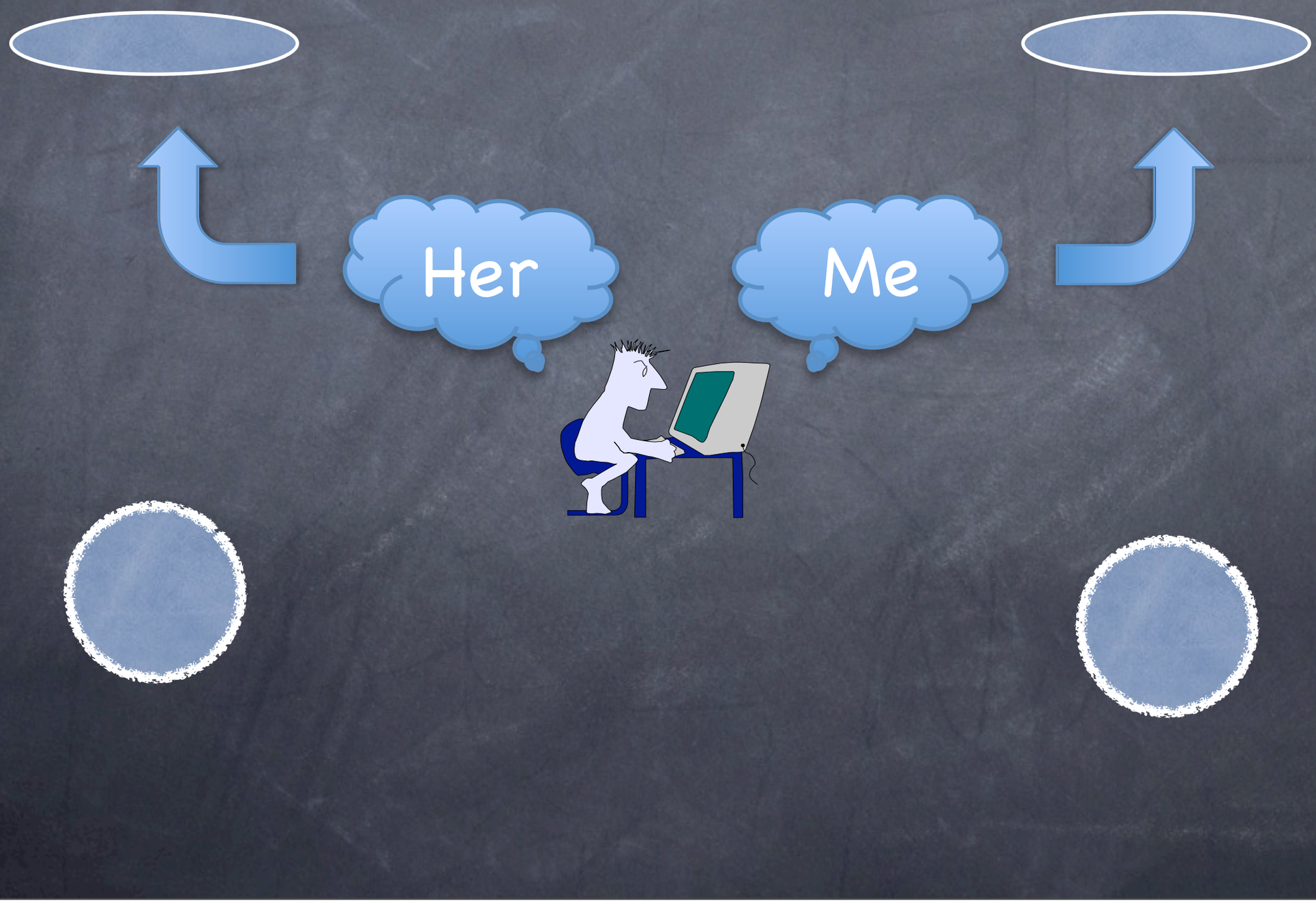
The human perspective
My phone, me, her, phone call

The human perspective



The human perspective
My phone, me, her, phone call

The human perspective

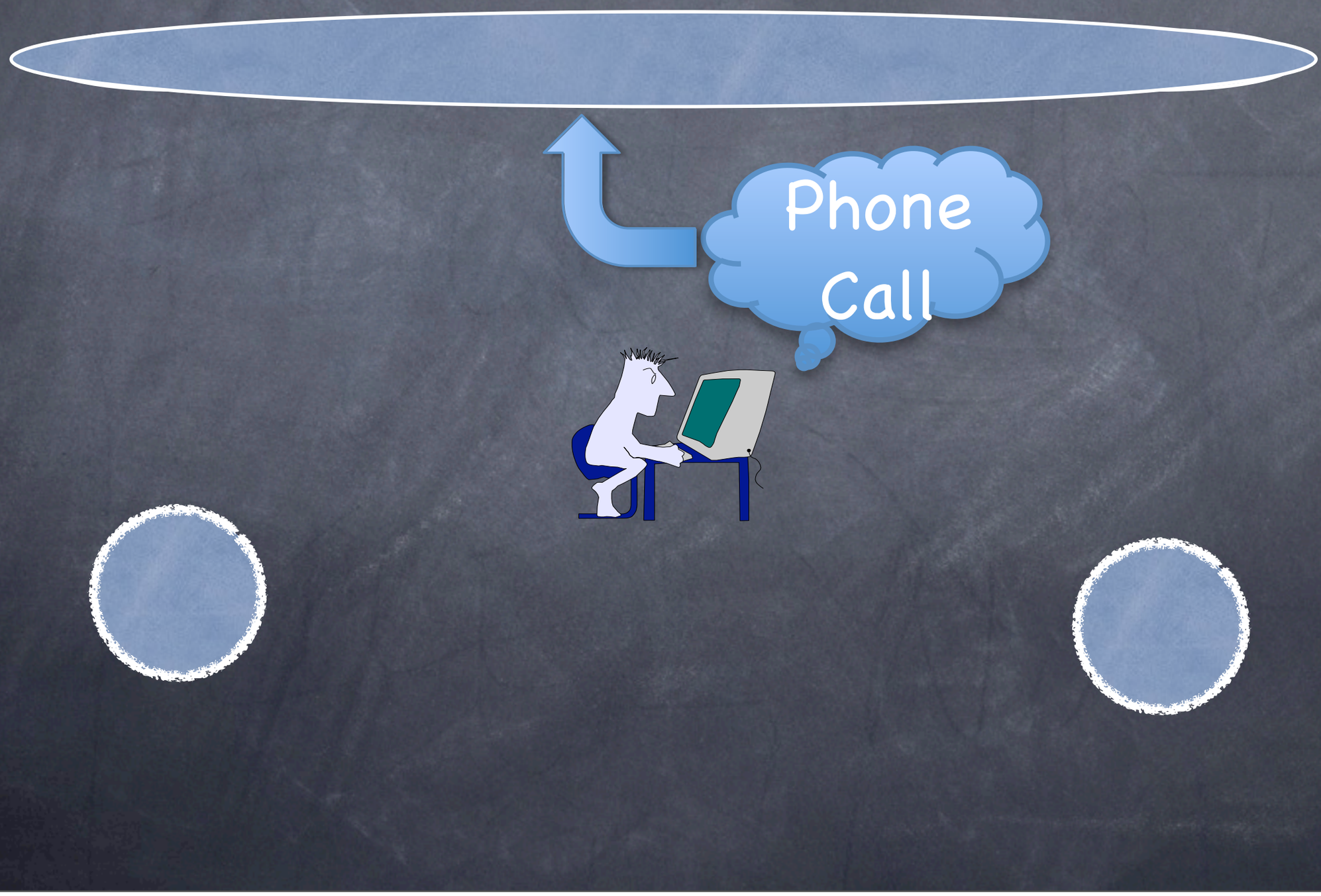


The human perspective
My phone, me, her, phone call

The human perspective

The human perspective
My phone, me, her, phone call

The human perspective



The human perspective
My phone, me, her, phone call

The human perspective

The human perspective
My phone, me, her, phone call

Symmetry breaking?

Living things, though often symmetrical, rarely have perfect symmetry. Indeed perfect symmetry is often a mark of death in things, rather than life. I believe the lack of clarity in the subject has arisen because of a failure to distinguish overall symmetry from local symmetries. ... In general, a large symmetry of the simplified neoclassicist type rarely contributes to the life of a thing, because in any complex whole in the world, there are nearly always complex, asymmetrical forces at work—matters of location, and context, and function—which require that symmetry be broken. — Alexander

Living things, though often symmetrical, rarely have perfect symmetry. Indeed perfect symmetry is often a mark of death in things, rather than life. I believe the lack of clarity in the subject has arisen because of a failure to distinguish overall symmetry from local symmetries. ... In general, a large symmetry of the simplified neoclassicist type rarely contributes to the life of a thing, because in any complex whole in the world, there are nearly always complex, asymmetrical forces at work—matters of location, and context, and function—which require that symmetry be broken. (Nature of Order, Book 1, p. 187)

Nature, too, creates beautiful structures which are governed by repeated application of structure-preserving transformations. In this connection, I think it is useful to remark that what I call structure-preserving transformations are very closely related to what has become known as "symmetry breaking" in physics.

— Alexander

Nature, too, creates beautiful structures which are governed by repeated appli- cation of structure-preserving transformations. In this connection, I think it is useful to remark that what I call structure-preserving transformations are very closely related to what has become known as "symmetry breaking" in physics. Christopher Alexander, Nature of Order, Book 1, pp. 63 – 64

The values of Lean Software Architecture

User and programmer sharing a mental model: Community
Domain models: Experience
Whole team: Deliberation
Real users: Soulful reflection
Just-in-time features: Incremental development
Piecemeal growth: Incremental Kaizen
Product: Geometry

The values of Lean Software Architecture

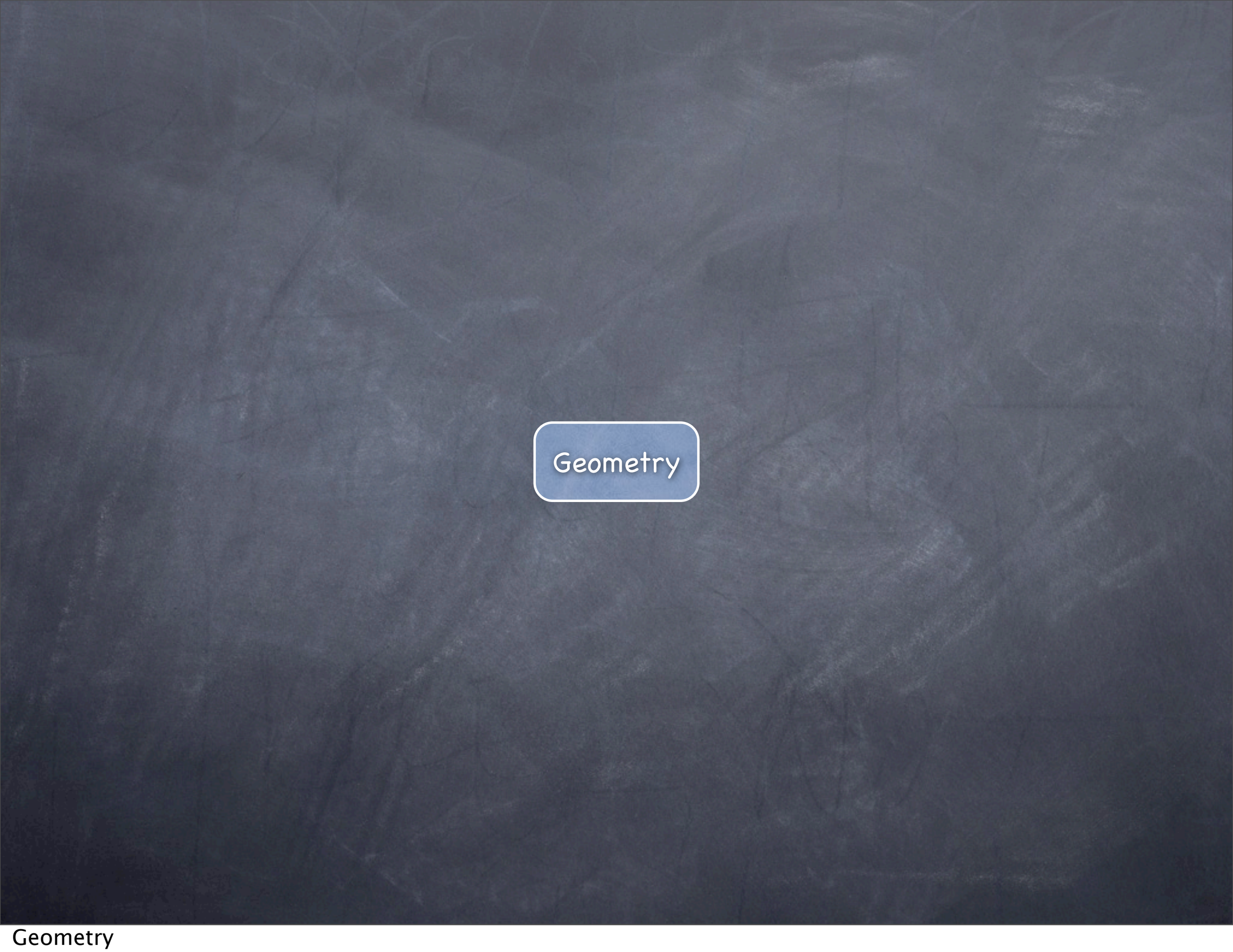
- User and programmer sharing a mental model
- Domain models
- Whole team
- Real users
- Just-in-time features
- Piecemeal growth
- Product

User and programmer sharing a mental model: Community
Domain models: Experience
Whole team: Deliberation
Real users: Soulful reflection
Just-in-time features: Incremental development
Piecemeal growth: Incremental Kaizen
Product: Geometry

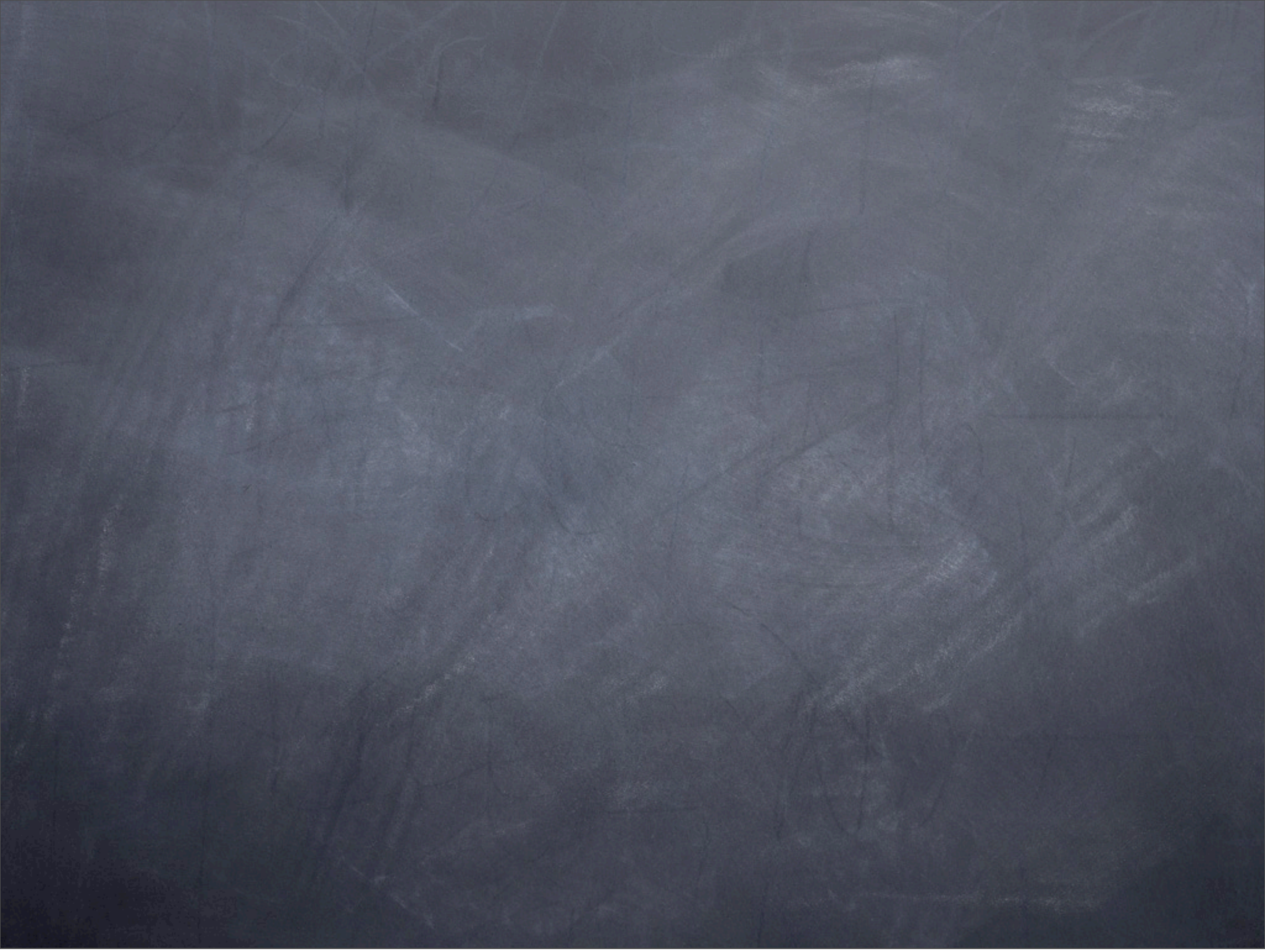
The values of Lean Software Architecture

Geometry

User and programmer sharing a mental model: Community
Domain models: Experience
Whole team: Deliberation
Real users: Soulful reflection
Just-in-time features: Incremental development
Piecemeal growth: Incremental Kaizen
Product: Geometry



Geometry



Geometry

Geometry

Geometry

Geometry

A pulsating, fluid, but nonetheless definite entity swims in your mind's eye. It is a geometrical image, it is far more than the knowledge of the problem; it is the knowledge of the problem, coupled with the knowledge of the kinds of geometrics which will solve the problem, and coupled with the feeling which is created by that kind of geometry solving that problem.

Geometry

A pulsating, fluid, but nonetheless definite entity swims in your mind's eye. It is a geometrical image, it is far more than the knowledge of the problem; it is the knowledge of the problem, coupled with the knowledge of the kinds of geometrics which will solve the problem, and coupled with the feeling which is created by that kind of geometry solving that problem.

Geometry

A pulsating, fluid, but nonetheless definite entity swims in your mind's eye. It is a geometrical image, it is far more than the knowledge of the problem; it is the knowledge of the problem, coupled with the knowledge of the kinds of geometrics which will solve the problem, and coupled with the feeling which is created by that kind of geometry solving that problem.

Alexander, The Timeless Way of Building, Chapter 9

Geometry

A pulsating, fluid, but nonetheless definite entity swims in your mind's eye. It is a geometrical image, it is far more than the knowledge of the problem; it is the knowledge of the problem, coupled with the knowledge of the kinds of geometrics which will solve the problem, and coupled with the feeling which is created by that kind of geometry solving that problem.

4. What are patterns about?

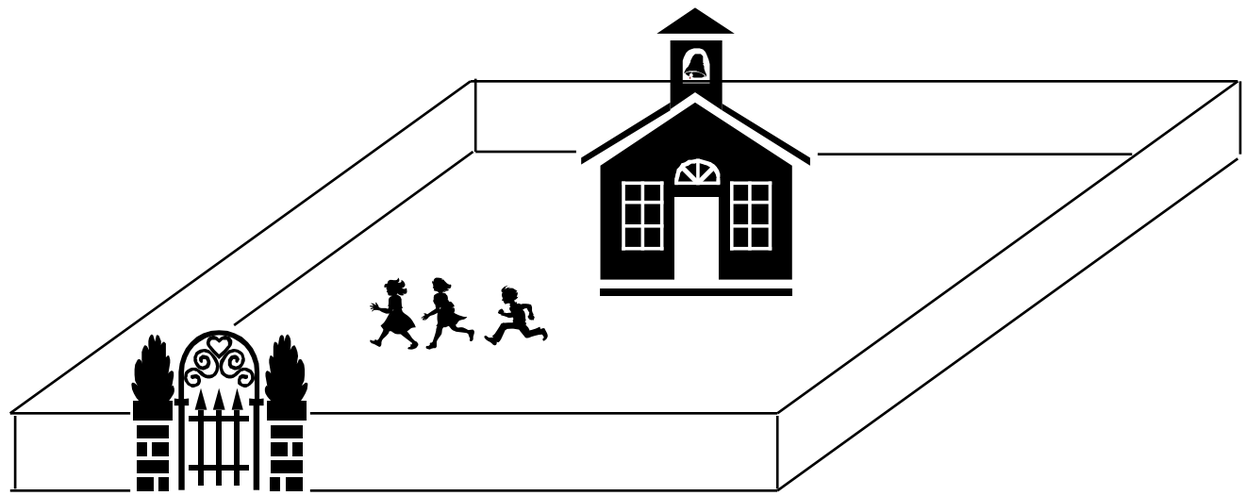
- Patterns compose into pattern languages
- A successful pattern language forms a paradigm or design style
  - Gothic Cathedral
  - DCI
- They are about product, wholeness, beauty, mu myu no shitsu — the One

4. What are patterns about?
Patterns compose into pattern languages
A successful pattern language forms a paradigm or design style
Gothic Cathedral
DCI
They are about product, wholeness, beauty, mu myu no shitsu — the One

風水

- Inexplicable common roots

  共通のルーツ



Foo shui / fung shui

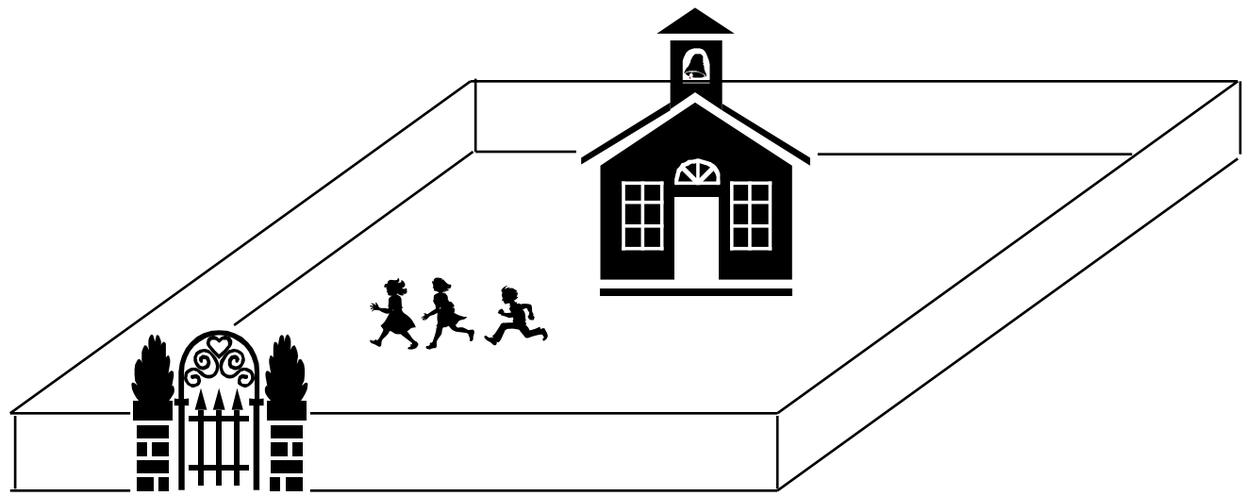
Nakano-san (Eishin school) and
Sasagawa-san

Common roots... why?

風水

- Inexplicable common roots

  共通のルーツ



何故ですか？

Foo shui / fung shui

Nakano-san (Eishin school) and
Sasagawa-san

Common roots... why?

Mu
myu no
shitsu

The
I

The
One

Summary

- Alexander's pattern values are similar to those of Lean
- Alexander's pattern values are almost absent from modern software patterns
- The values live on in Scrum and Lean Architecture
- Software patterns should revisit their Alexandrian roots
- Software patterns can learn from Lean Architecture
- As ten years ago: chances for success in Japan is higher than in the West

Alexander's pattern values are similar to those of Lean
Alexander's pattern values are almost absent from modern software patterns
The values live on in Scrum and Lean Architecture
Software patterns should revisit their Alexandrian roots
Software patterns can learn from Lean Architecture
As ten years ago: chances for proper adoption in Japan is higher than in the West