Guiding Diagrams

We've equipped wiki with a means to draw pictures of work in progress that can guide this work to completion.

Guiding Emergent Structure with Diagrams
April 11-13, 2019 • University of Oregon Portland
What is Technology? post ⧉

Showcase

Michael Mehaffy's Growing Regions is a refresh of a classic pattern language meant to relaunch a movement.

Tree Bressen's Categories organizing a pattern language for groups printed on a deck of playing cards.

Thompson Morrison's Dayton Experiment emergent pattern language based on five years reinventing education.

Background

El Dorado used diagrams to facilitate browsing an enterprise scale graph database.

More About Algorithmic Markup describes the language we designed for drawing knowledge embedded in wiki.
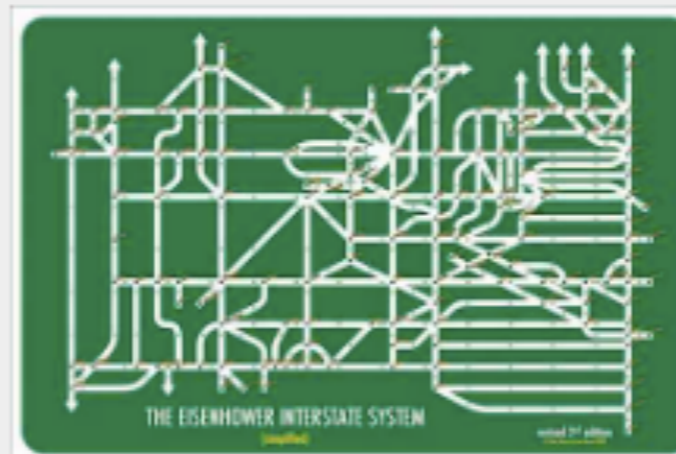
find pages in
gd.fed.wiki

History

Visualizing High-Order Graphs

How might we draw meaningful diagrams of wiki that don't turn quickly into a fuzzball of links?

We desire to see more of wiki all at once. We might use a graph to consider where we are, where we might go, and how we should get there in the wiki's information space.

The Interstate Highway map provides an ideal where hundreds of nodes are placed in meaningful relation and connected with often two but sometimes ten arcs each.
source

The dense linking in wiki works against the notion of meaningful maps. There is no obvious spatial relationship to guide node placement. Nor is there room for all of the arcs between nodes. There are too many to be represented with lines.
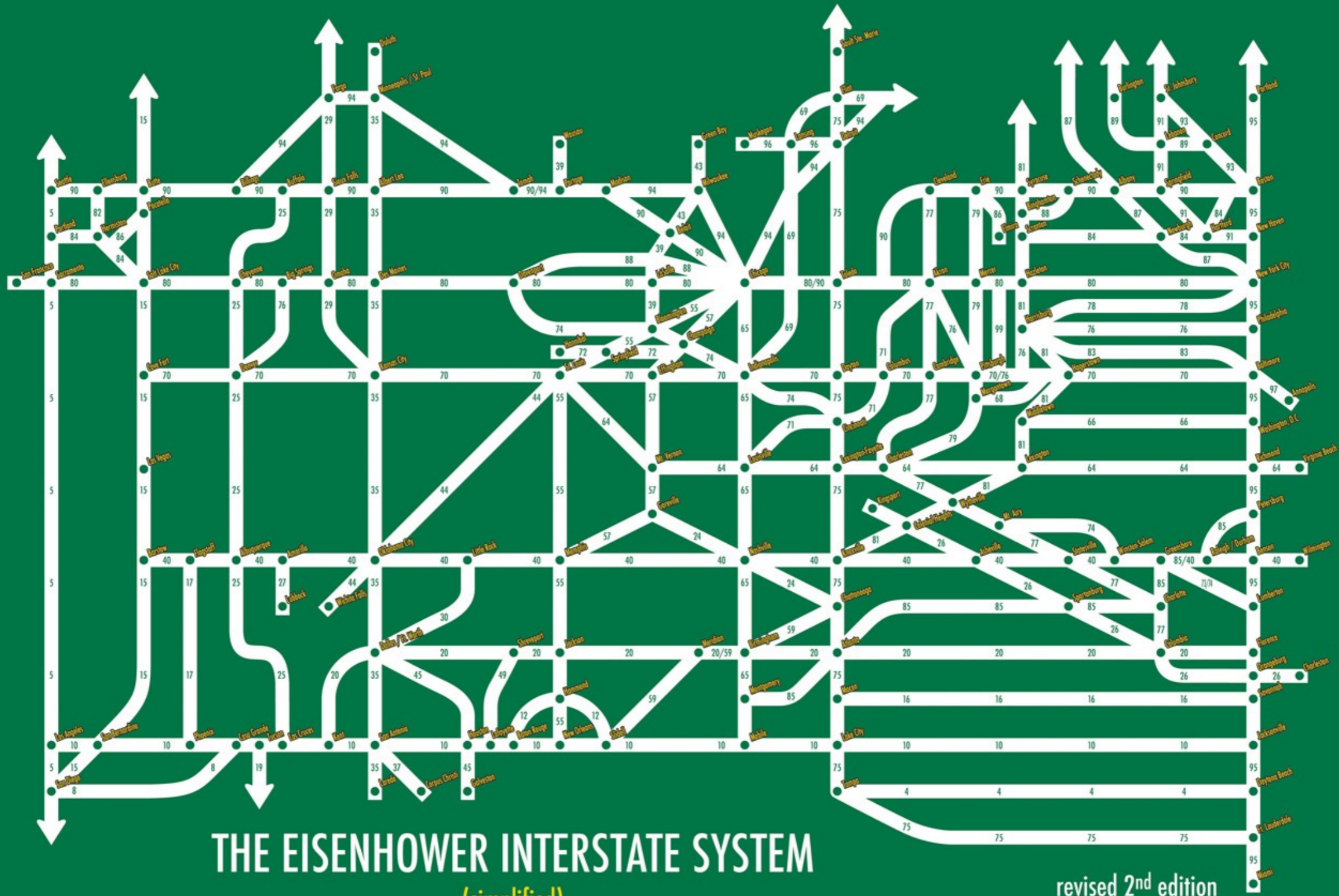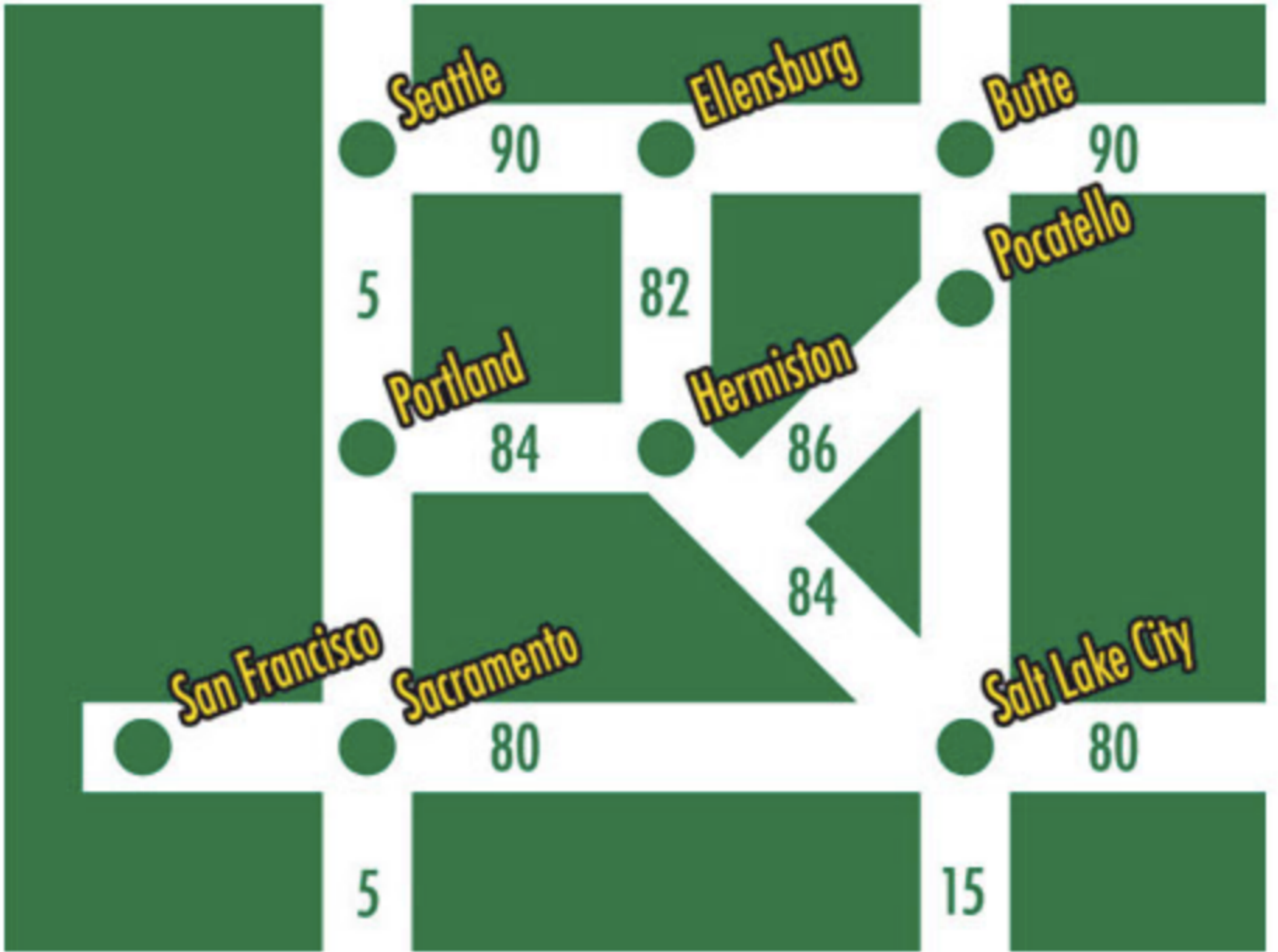
THE EISENHOWER INTERSTATE SYSTEM
(simplified)

revised 2nd edition
© Chris Yates & Dan Burns 2009

≡ **Google** Scholar 🔍

Stephen North ✉ FOLLOW

graphviz.org, Infovisible, assoc editor IEEE TVCG, ex-AT&T Fellow
Verified email at acm.org - Homepage

visualization   applied algorithms   data analysis   code

ARTICLES	CITED BY	CO-AUTHORS

TITLE	CITED BY	YEAR
CorrelatedMultiples: Spatially Coherent Small Multiples With Constrained Multi-Dimensional Scaling X Liu, Y Hu, S North, HW Shen Computer Graphics Forum 37 (1), 7-18	20	2018
What you see is what you can change: Human-centered machine learning by interactive visualization D Sacha, M Sedlmair, L Zhang, JA Lee, J Peltonen, D Weiskopf, SC North, ... Neurocomputing 268, 164-175	33	2017
Visual interaction with dimensionality reduction: A structured literature analysis D Sacha, L Zhang, M Sedlmair, JA Lee, J Peltonen, D Weiskopf, SC North, ... IEEE transactions on visualization and computer graphics 23 (1), 241-250	71	2017
Human-centered machine learning through interactive visualization D Sacha, M Sedlmair, L Zhang, JA Lee, D Weiskopf, S North, D Keim ESANN	33	2016

2000

This is my first attempt to use GraphViz to trace links on wiki. Here I chose to follow the links to the two largest pages cited on any given page. I'm thinking the two most recently edited pages might be an even better choice.
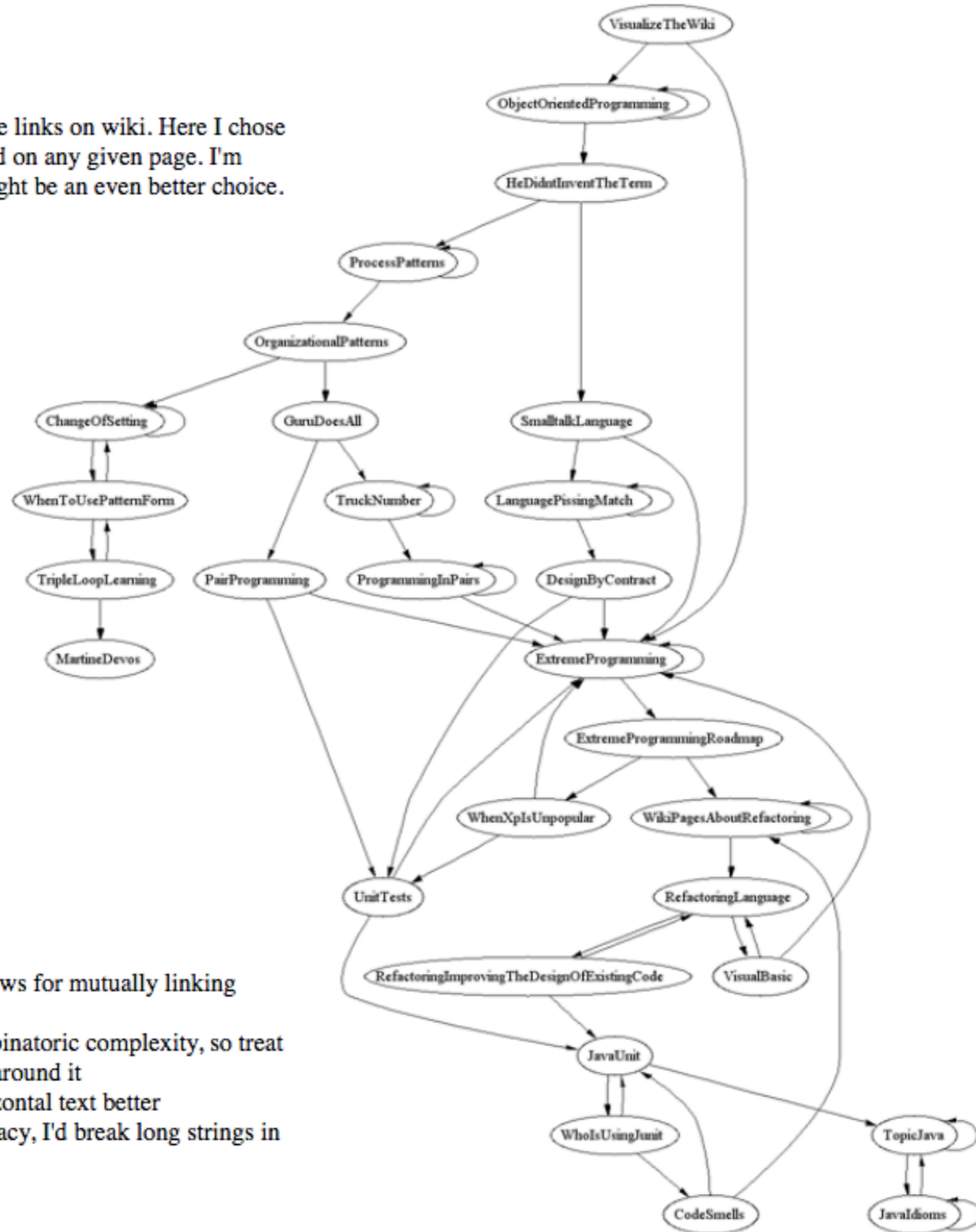
Hi Ward. Here's some suggestions:

- Use double-headed arrows instead of two arrows for mutually linking pages
- Drop the references to self - they add no combinatoric complexity, so treat them as attributes of a node, not of the edges around it
- Rectangular nodes ("box" attribute) treat horizontal text better
- If I had a UserStory for readability over accuracy, I'd break long strings in the middle and double-deck them

Graph a Structure with Graphviz

Graphviz will draw a graph composed of nodes and arrows from a declarative text description. Here we describe how to get something up fast and then annotate that with useful debugging results.

Things to Try

Make a file, try1.dot, that contains this dot program. Open it with Graphviz. download ⧉

```
digraph {
   1 -> 2 -> 3 -> 4;
   1 -> 3;
   2 -> 4;
}
```
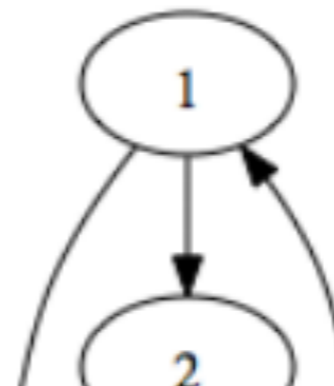
Try making a cycle by adding an edge bottom to top.

```
   4 -> 1;
```

How it Works

Nodes

A node can be a number or a name or an arbitrary string inside of quotes.

Dump a Structure into Graphviz

Many algorithms represent graphs or trees. Use Graphviz to check that these structures are what you think they are. Examples in Ruby.

First try Graph a Structure with Graphviz

Things to Try

Create an array variable to hold dot statements generated as the program runs. Add defaults for nodes and edges.

```
@dot = []
@dot << "node [style=filled fillcolor=lightBlue];"
```

Add nodes and edges as your program runs. Write a subroutine to traverse your data structures if this helps.

```
@dot << "#{quote a} -> #{quote b};"
```

Define a function to add quotes around any string you use for node names.

```
def quote string
    "\"#{string}\""
end
```

Write the dot file at the end of the run.
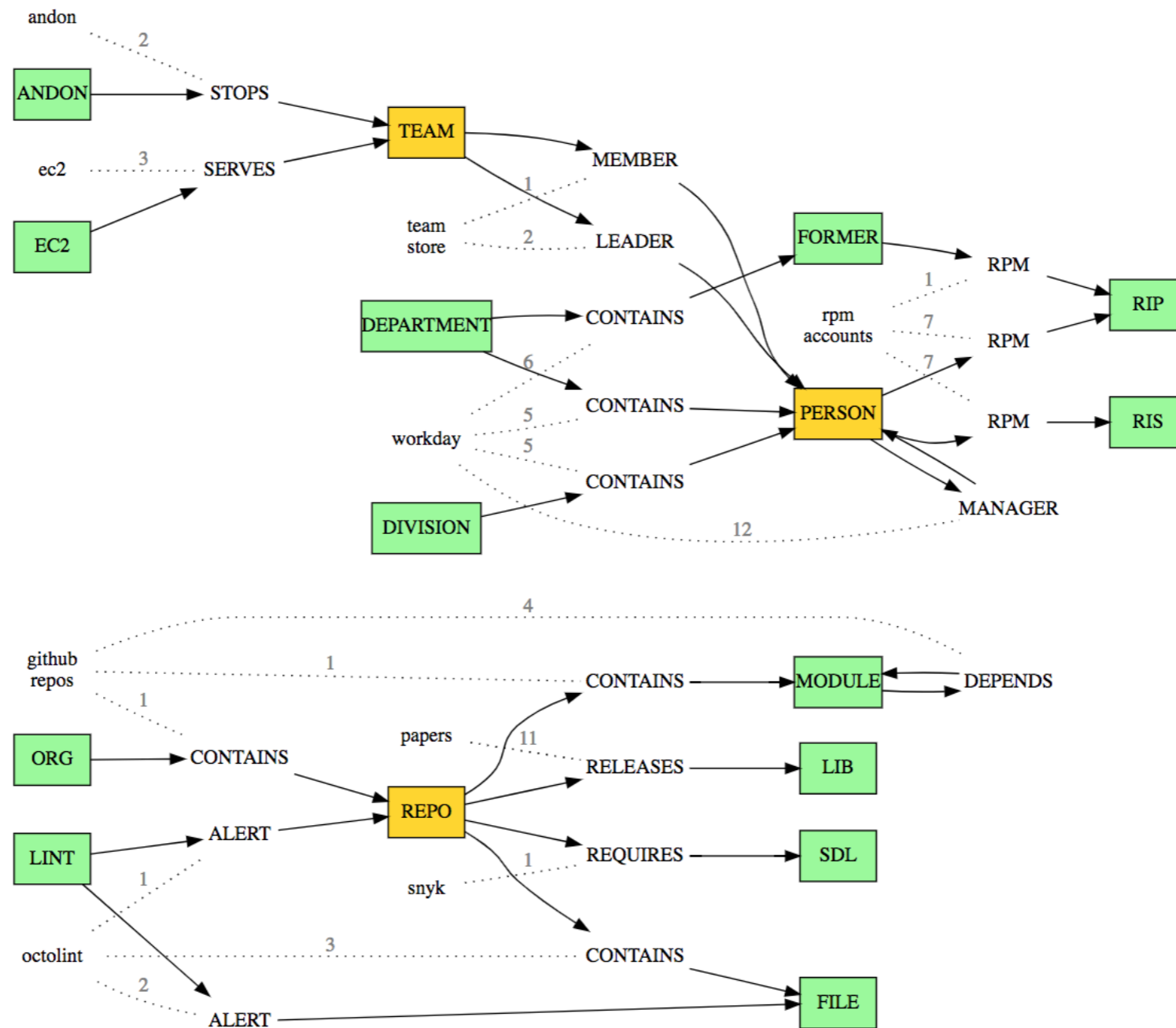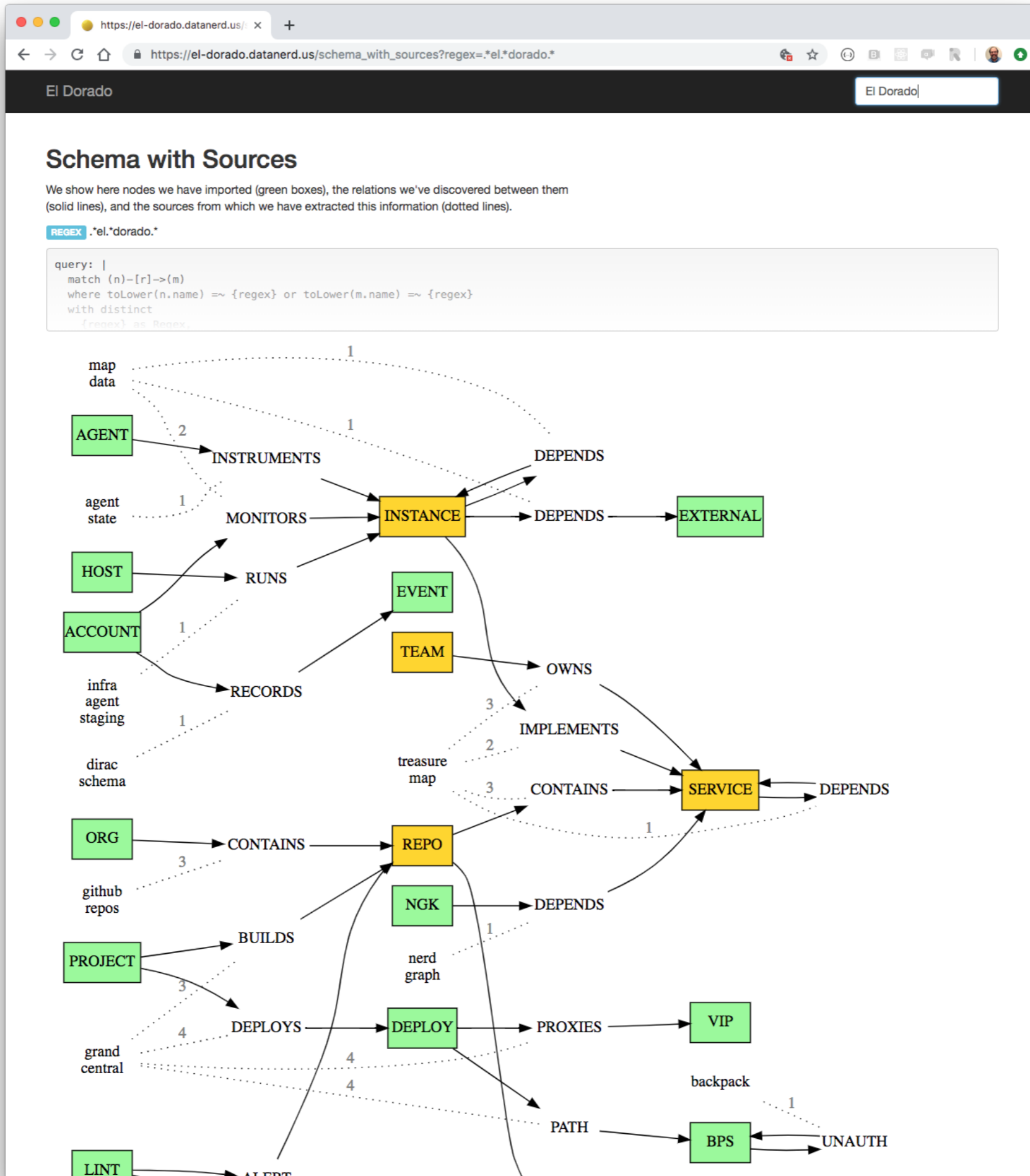
Schema with Sources

We show here nodes we have imported (green boxes), the relations we've discovered between them (solid lines), and the sources from which we have extracted this information (dotted lines).

REGEX .*el.*dorado.*

```
query: |
  match (n)-[r]->(m)
  where toLower(n.name) =~ {regex} or toLower(m.name) =~ {regex}
  with distinct
    {regex} as Regex,
```

Patterns

◼ Common Areas At The Heart

. . . along the Intimacy Gradient, in every building and in every social group within the building, it is necessary to place the common areas. Place them on the sunlit side to reinforce the pattern of Indoor Sunlight; and, when they are large, give them the higher roofs of the CASCADE OF ROOFS (116).

No social group - whether a family, a work group, or a school group - can survive without constant informal contact among its members.

Therefore:

Create a single common area for every social group. Locate it at the center of gravity of all the spaces the group occupies, and in such a way that the paths which go in and out of the building lie tangent to it.

❖ ❖ ❖

Any building which houses a

center of gravity of social life

🟪 Light On Two Sides Of Every Room

. . . once the building's major rooms are in position, we have to fix its actual shape: and this we do essentially with the position of the edge. The edge has got its rough position already from the overall form of the building - Wings Oflight, Positive Outdoor Space, Long Thin House, Cascade Of Roofs. This pattern now completes the work of Wings Oflight, by placing each individual room exactly where it needs to be to get the light. It forms the exact line of the building edge, according to the position of these individual rooms. The next pattern starts to shape the edge.

When they have a choice, people will always gravitate to those rooms which have light on two sides, and leave the rooms which are lit only from one side unused and empty.

◼ Window Place

. . . this pattern helps complete the arrangement of the windows given by Entrance Room, Zen View, Light On Two Sides Of Every Room, Street Windows. According to the pattern, at least one of the windows in each room needs to be shaped in such a way as to increase its usefulness as a space.
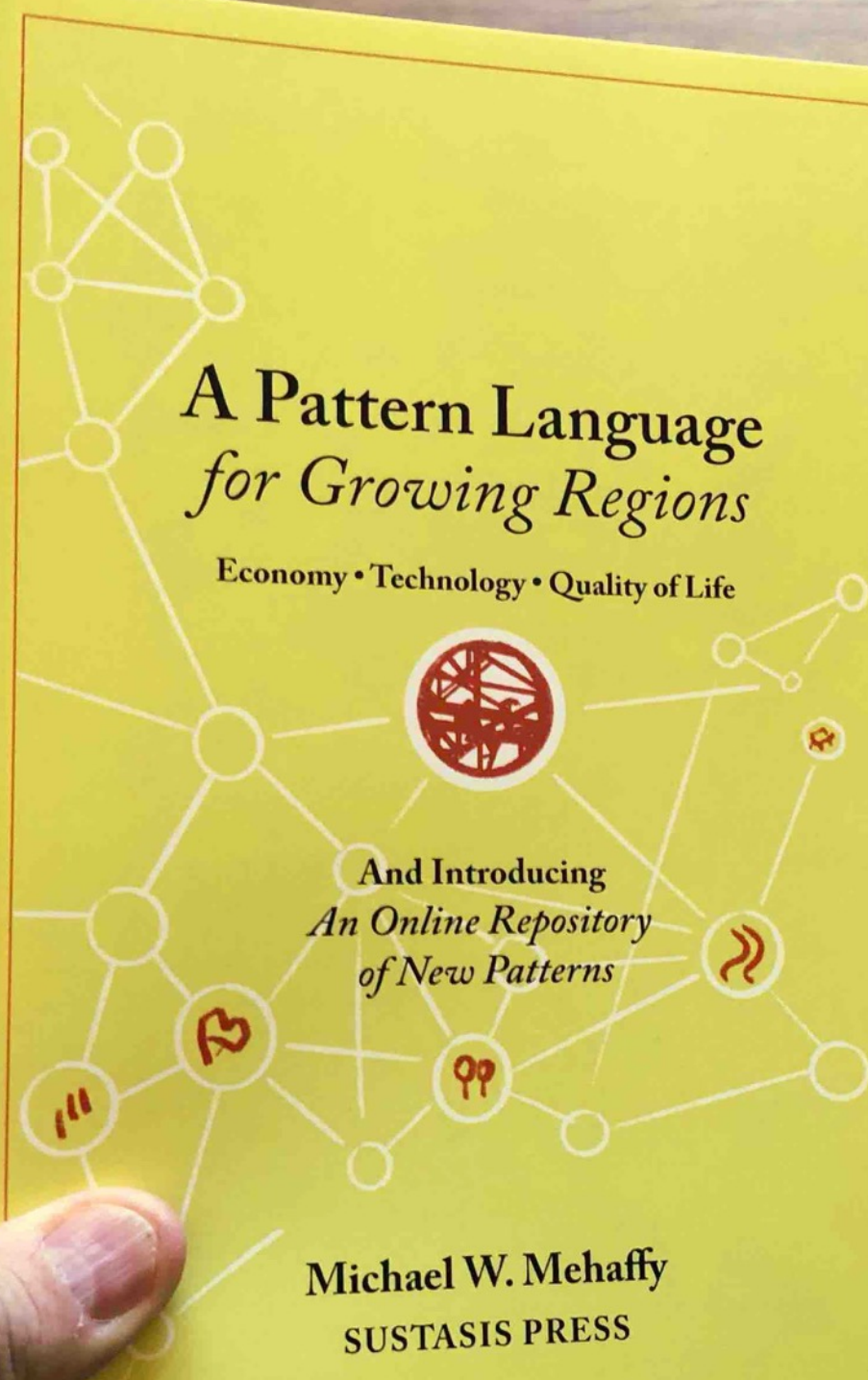
Everybody loves window seats, bay windows, and big windows with low sills and comfortable chairs drawn up to them.

Therefore:

In every room where you spend any length of time during the day, make at least one window into a "window place."

```
Gradients
and
connection
of
space

Common
Areas
At
The
Heart

Light
On
Two
Sides
Of
Every
Room

Window
Place

Short Passages
Outdoor Room
Arcades
Positive Outdoor Space
Long Thin House
Deep Reveals
Built-in Seats
Dormer Windows
Street Windows
Alcoves
Low Sill
Cascade Of Roofs
Roof Layout
Farmhouse Kitchen
Communal Eating
Windows Overlooking Life
Filtered Light
The Fire
Natural Doors And Windows
The Shape Of Indoor Space
Intimacy Gradient
Number Of Stories
Main Entrance
A Room Of One's Own
```

A Pattern Language
for Growing Regions

Economy · Technology · Quality of Life

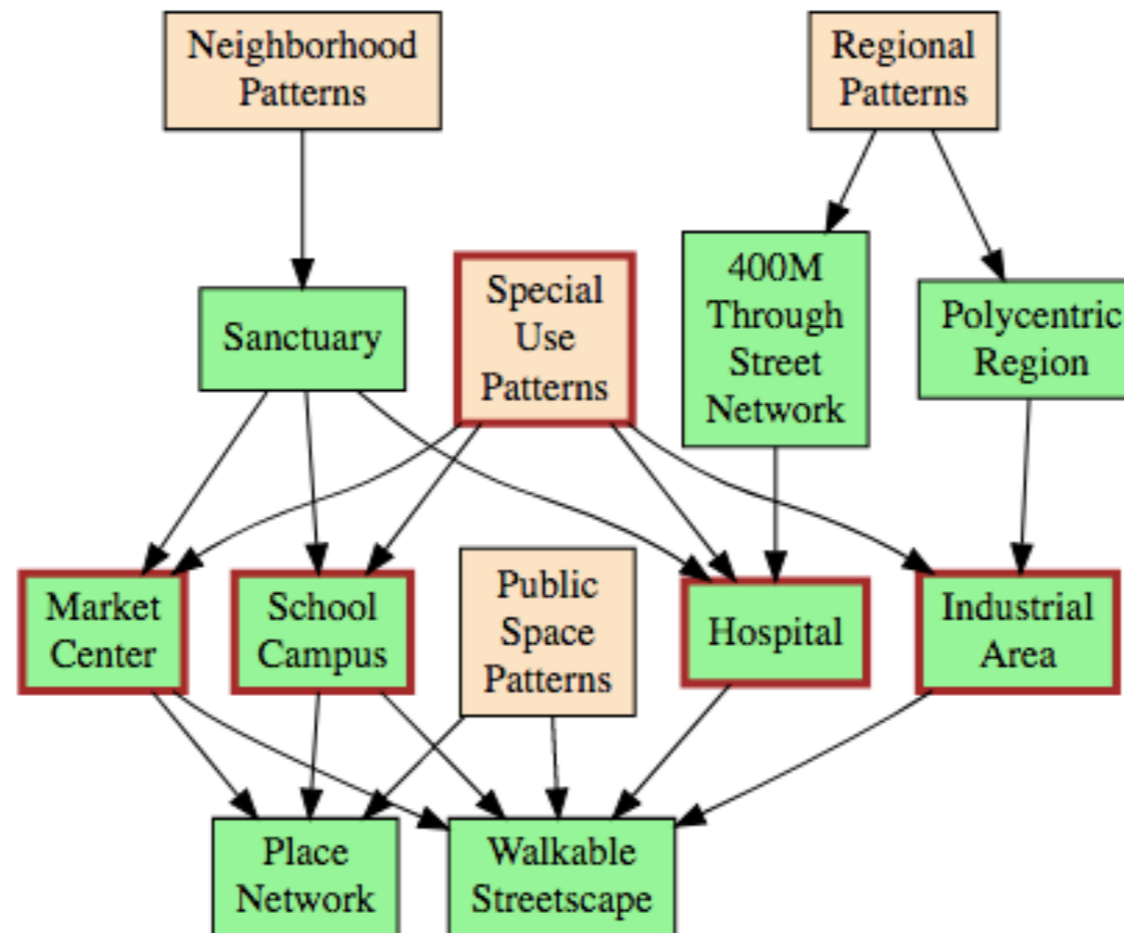**And Introducing**
*An Online Repository
of New Patterns*

Michael W. Mehaffy

SUSTASIS PRESS

Special Use Patterns
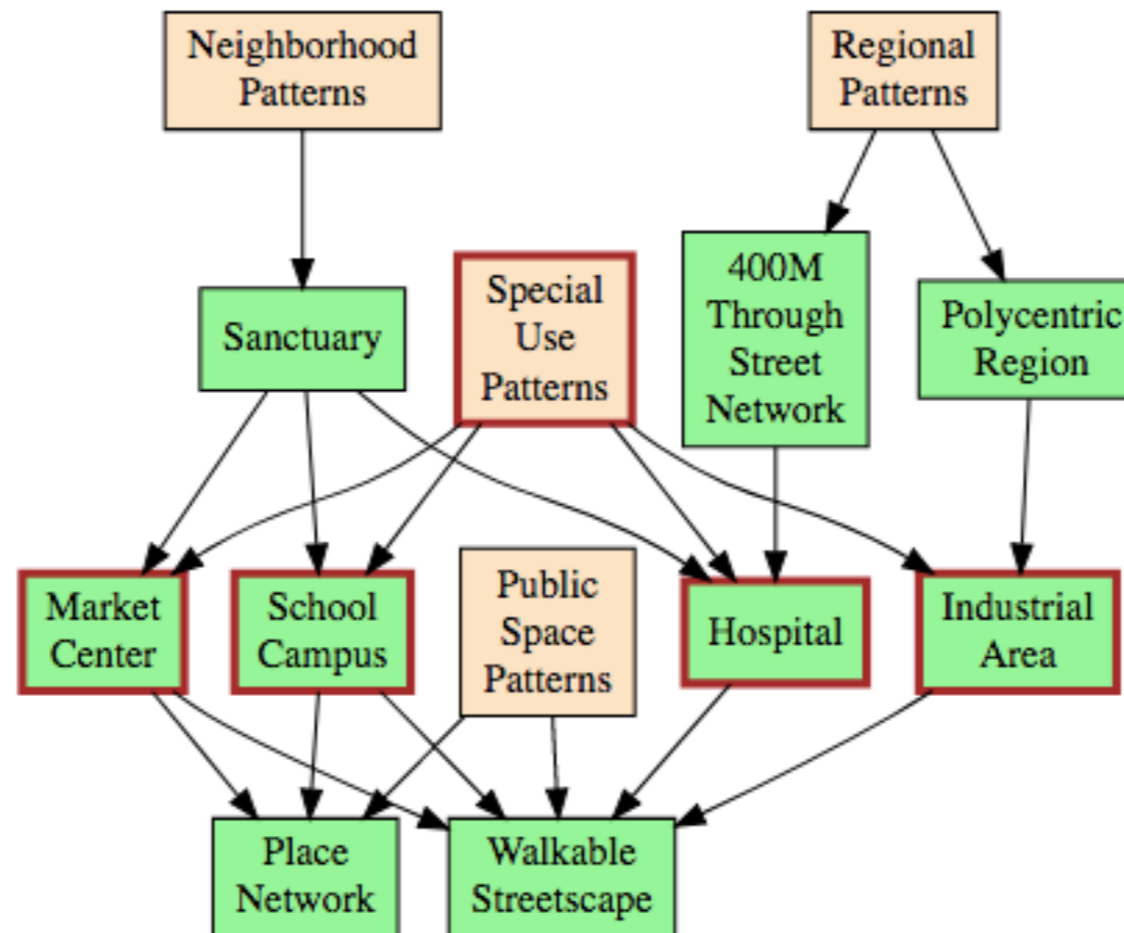
What force unites these patterns?

- School Campus
- Market Center
- Industrial Area
- Hospital

Public Space Patterns

What force unites these patterns?

```
Neighborhood                                   Regional
 Patterns                                      Patterns

                                                    400M
                    Special                        Through         Polycentric
   Sanctuary         Use                            Street          Region
                   Patterns                        Network

   Market    School    Public                              Industrial
   Center    Campus    Space      Hospital                   Area
                       Patterns

         Place        Walkable
        Network      Streetscape
```

- School Campus
- Market Center
- Industrial Area
- Hospital

■ Walkable Streetscape

Along streets, pedestrians need ample space to walk, and to pass other pedestrians who are walking i

There are many potential conflicts between pedestrians and other forms of movement, as well as potential conflicts between pedestrian needs and the needs of adjacent building users.

Therefore:

Assure that every streetscape along an avenue, greenway corridor or multi-way boulevard is walkable, by providing adequate width for pedestrian travel (typically at least 3 meters, or 10 feet) in addition to space for seating, and space for planting and light poles.

When …Along the Avenue (p. 58), Greenway Corridor (p. 52) and Multi-Way Boulevard (p. 55) there is a need to provide for Walkable Multi-MobilitY (p. 38)...

Then Line the streetscape wherever possible with Perimeter Building (p. 116), and elsewhere place pedestrian-friendly visual elements such as trellises, pergolas, vegetation and other attractive screens. Provide Human-Scale Detail (p. 154)including architectural and urban elements, along the length of the streetscape ...

See more Public Space Patterns
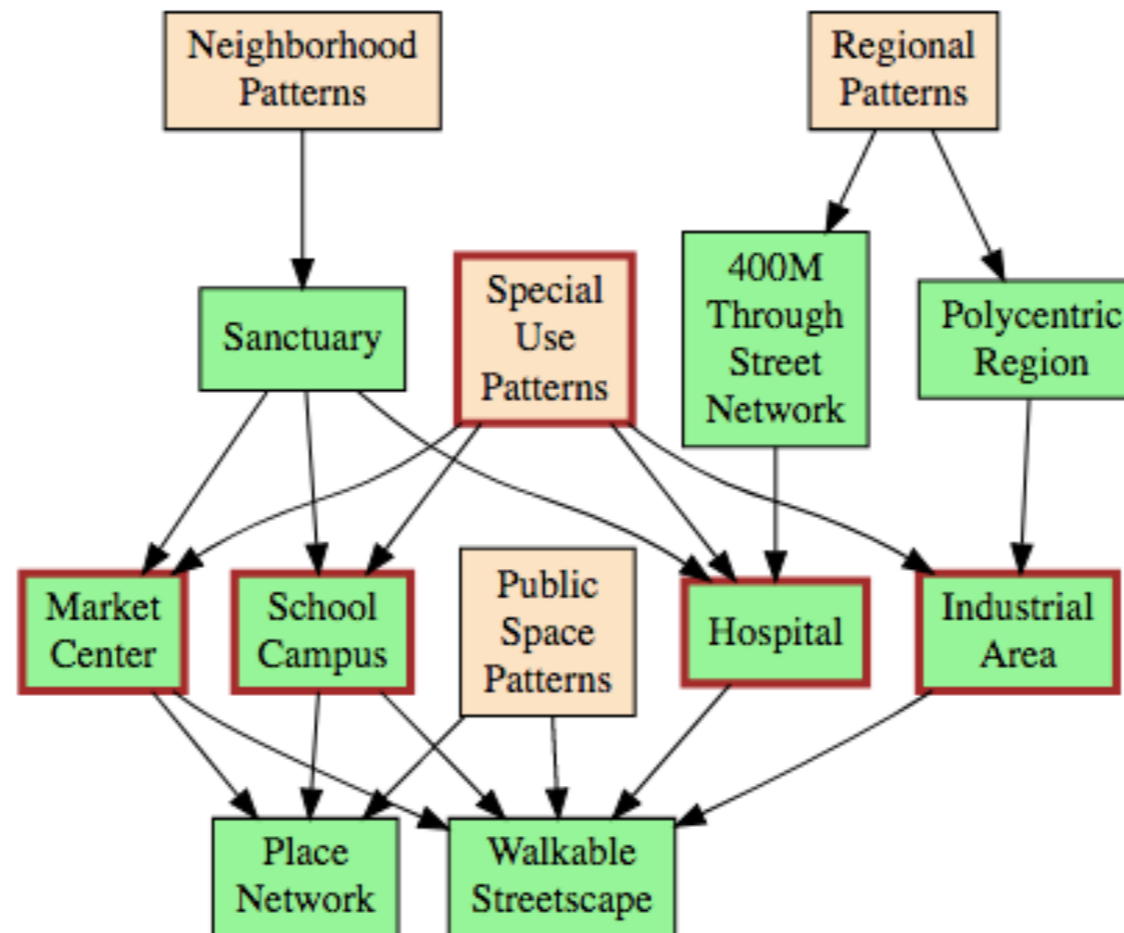
- School Campus
- Market Center
- Industrial Area
- Hospital

Emergence

The Dayton Experiment

It started as an experiment. We called it the Dayton Experiment.

What might happen if you take the learning culture that fuels our fastest growing high technology companies and used that culture to re-imagine the educational experience in a school district?
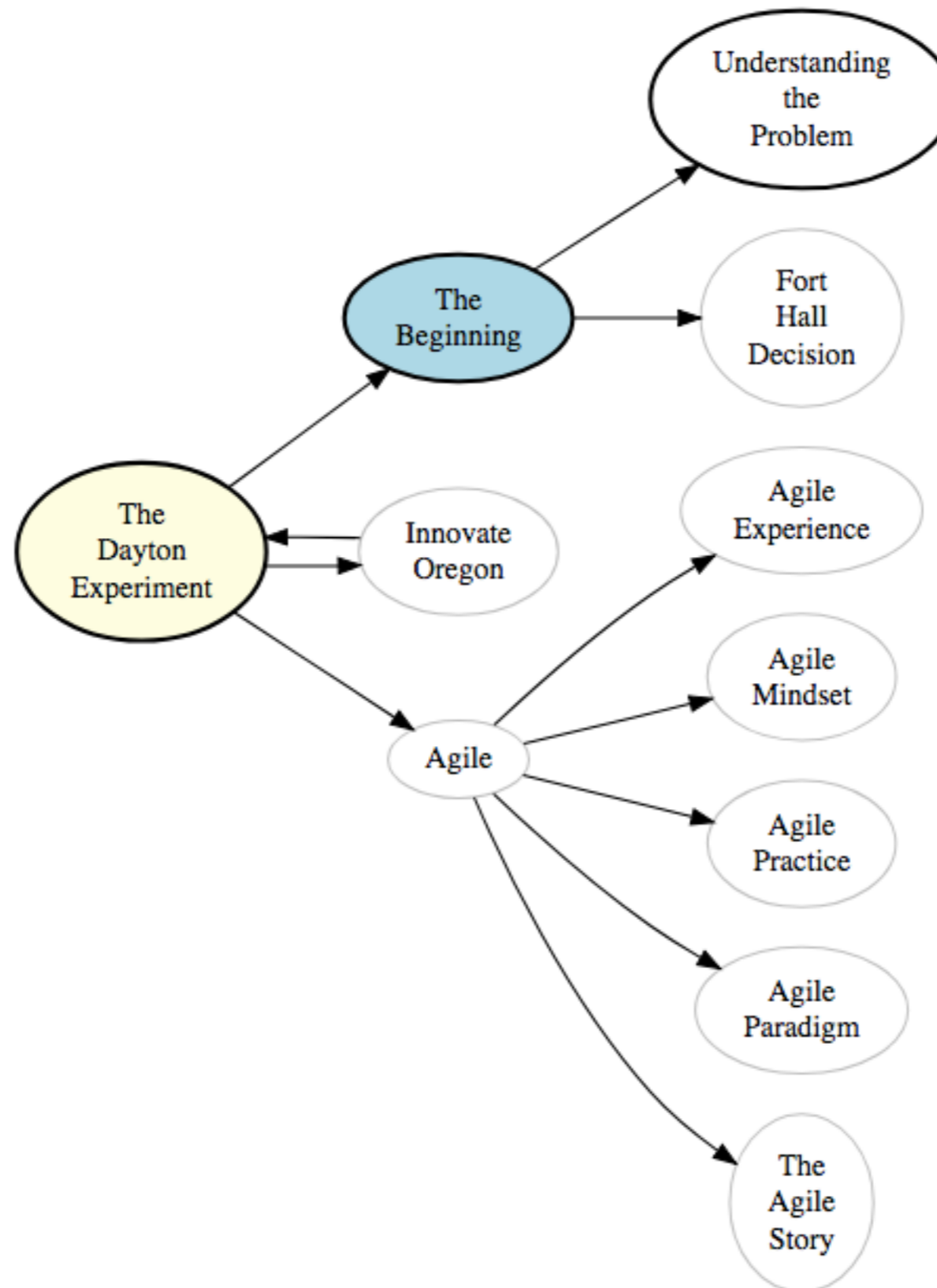
In 2015, under the banner of Innovate Oregon, I embarked on this adventure in Dayton Oregon, a rural community in Yamhill County. Dayton is a town of 2,500 people where more than two-thirds of its students live near the poverty line. It is also a multicultural community, about a third of its students are Latinx.
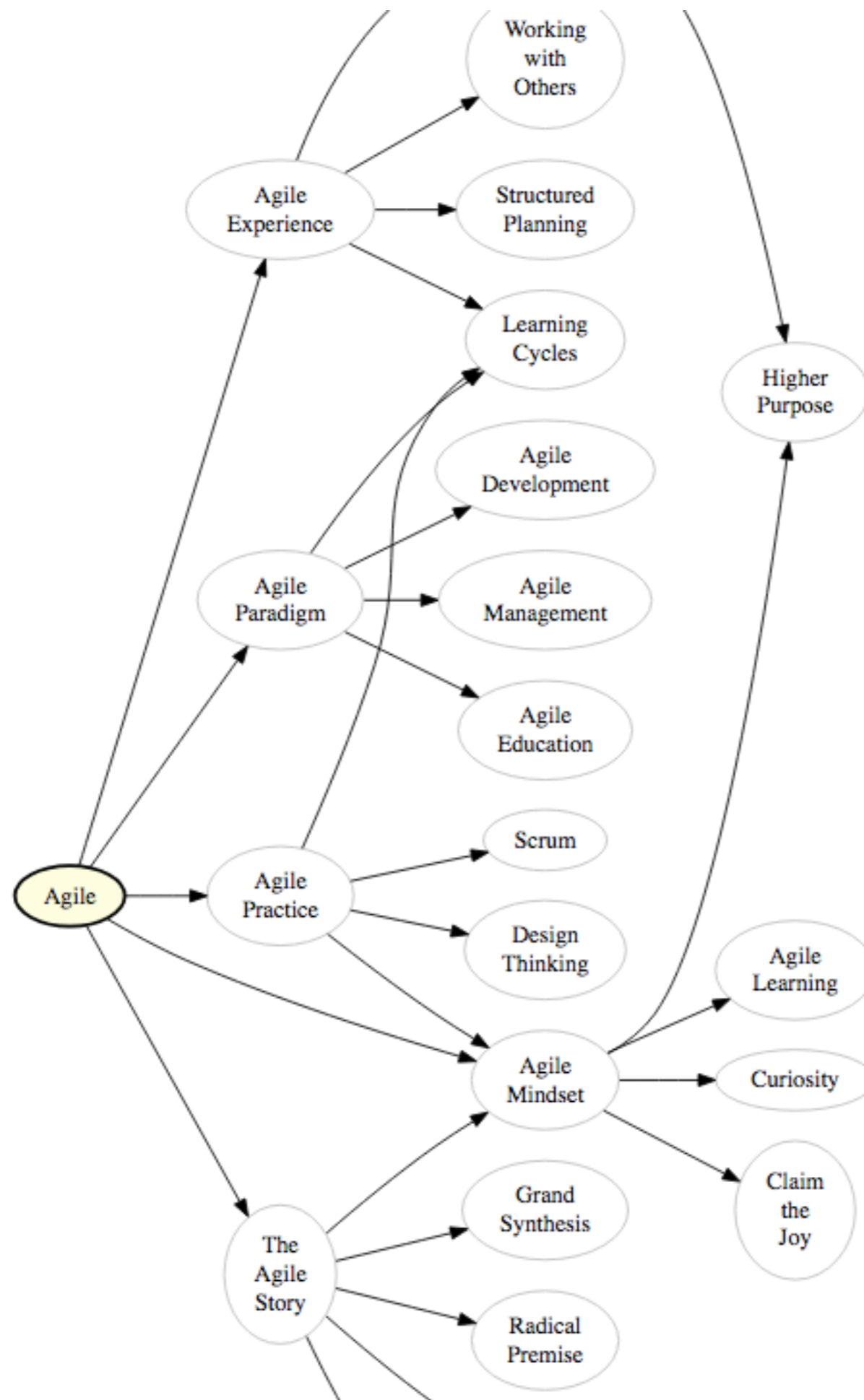
We call this new culture Agile. It was birthed in 2001 as a movement and continues to transform how software is being developed and how companies are being managed. Could it also transform how we are preparing the next generation of students to be the makers in a new creative economy?
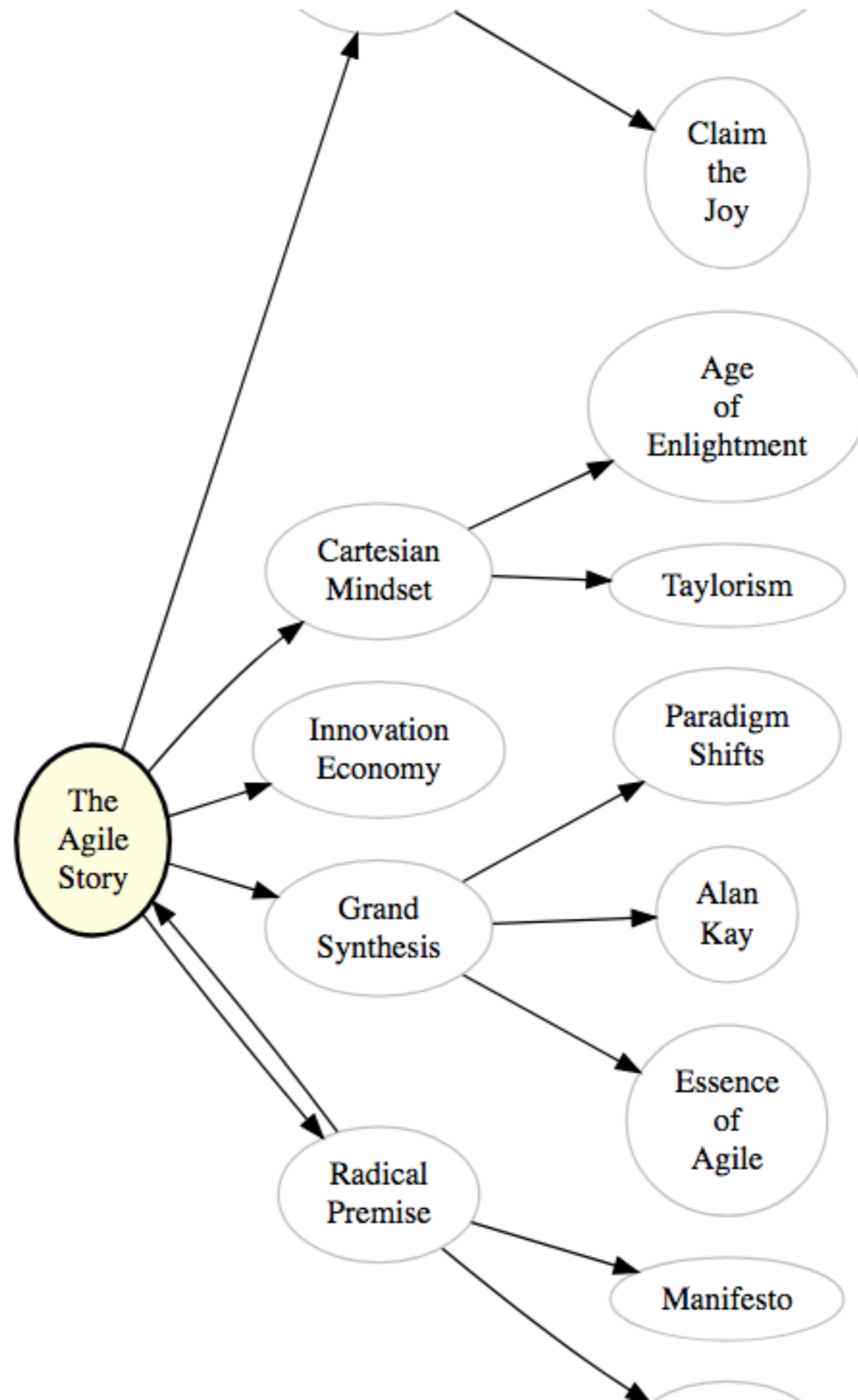
The experiment has blossomed beyond anyone's expectations. Dayton's graduation rate is now one of the highest in the state, 97%, 18% above the state average. The town has a 10 Gig fiber internet network, the fastest on the entire west coast, that was built by a committed

And they are becoming an inspirational model for other districts around the state and beyond.

It's a story worth telling.

Agile

Agile Experience
- Working with Others
- Structured Planning
- Learning Cycles

Agile Paradigm
- Learning Cycles
- Agile Development
- Agile Management
- Agile Education

Agile Practice
- Scrum
- Design Thinking

Higher Purpose

Agile Mindset
- Agile Learning
- Curiosity
- Claim the Joy

The Agile Story
- Grand Synthesis
- Radical Premise

The Agile Story

- Claim the Joy
- Cartesian Mindset
  - Age of Enlightment
  - Taylorism
- Innovation Economy
- Grand Synthesis
  - Paradigm Shifts
  - Alan Kay
  - Essence of Agile
- Radical Premise
  - Manifesto
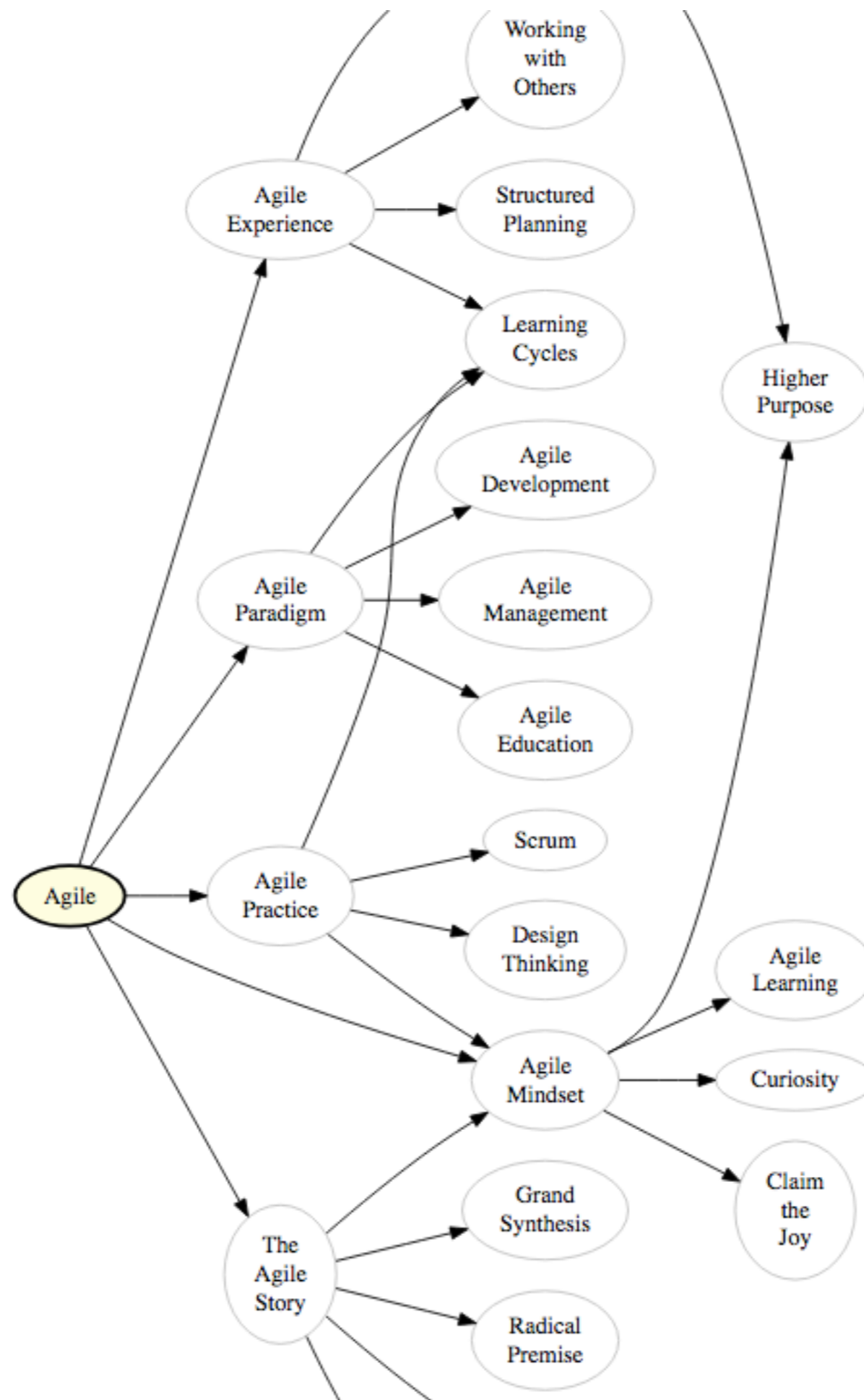
Ward Cunningham <ward@c2.com>
To: Thompson Morrison <thompson@innovateoregon.org>
Agile in four paragraphs?

March 22, 2019 at 7:52 PM
Sent - c2.com

Thompson — I wonder if you could write a new Agile page in four paragraphs each with its own link to 1/4 of these nodes? I'll help if you like. — Ward

Ok, I refactored the page for you.

http://path.ward.asia.wiki.org/agile.html

I did this by naming the four links, two of which already existed. Then dragging paragraphs from the one page to the others where they seemed to fit in. Then I deleted the original Agile page and started over with four paragraphs each in the same form with the four links in the same sentence structure so it was clear that I was offering alternative paths.

The Agile Experience where commonly faced problems find fresh solutions that are hard to understand but easy to experience.

The Agile Mindset where one learns to look for similarly fresh solutions in diverse activities.

The Agile Practice where specific solutions have been assembled into a framework supported by available training and coaching.

The Agile Paradigm where recognizably similar solutions are adopted by distinctly different activities.

This has flattened out the associated graph which is now readable within the confines of the wiki page. Agile Experience might be asking for the same treatment now.
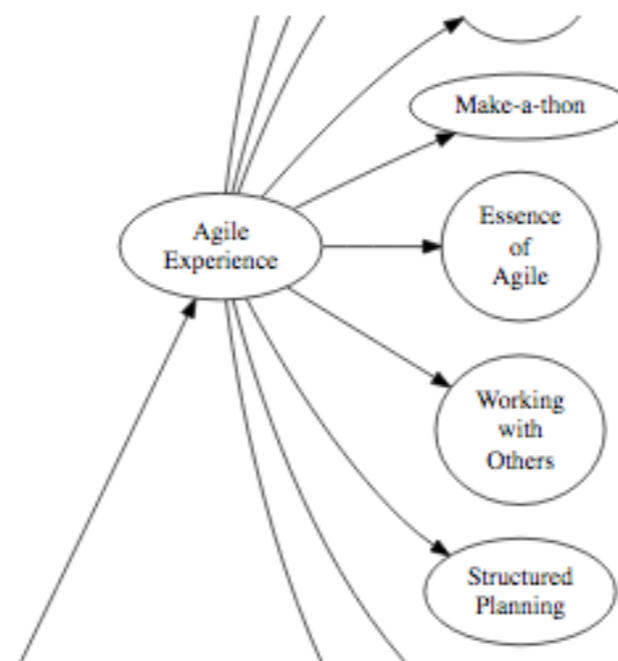
And they are becoming an inspirational model for other districts around the state and beyond.

It's a story worth telling.
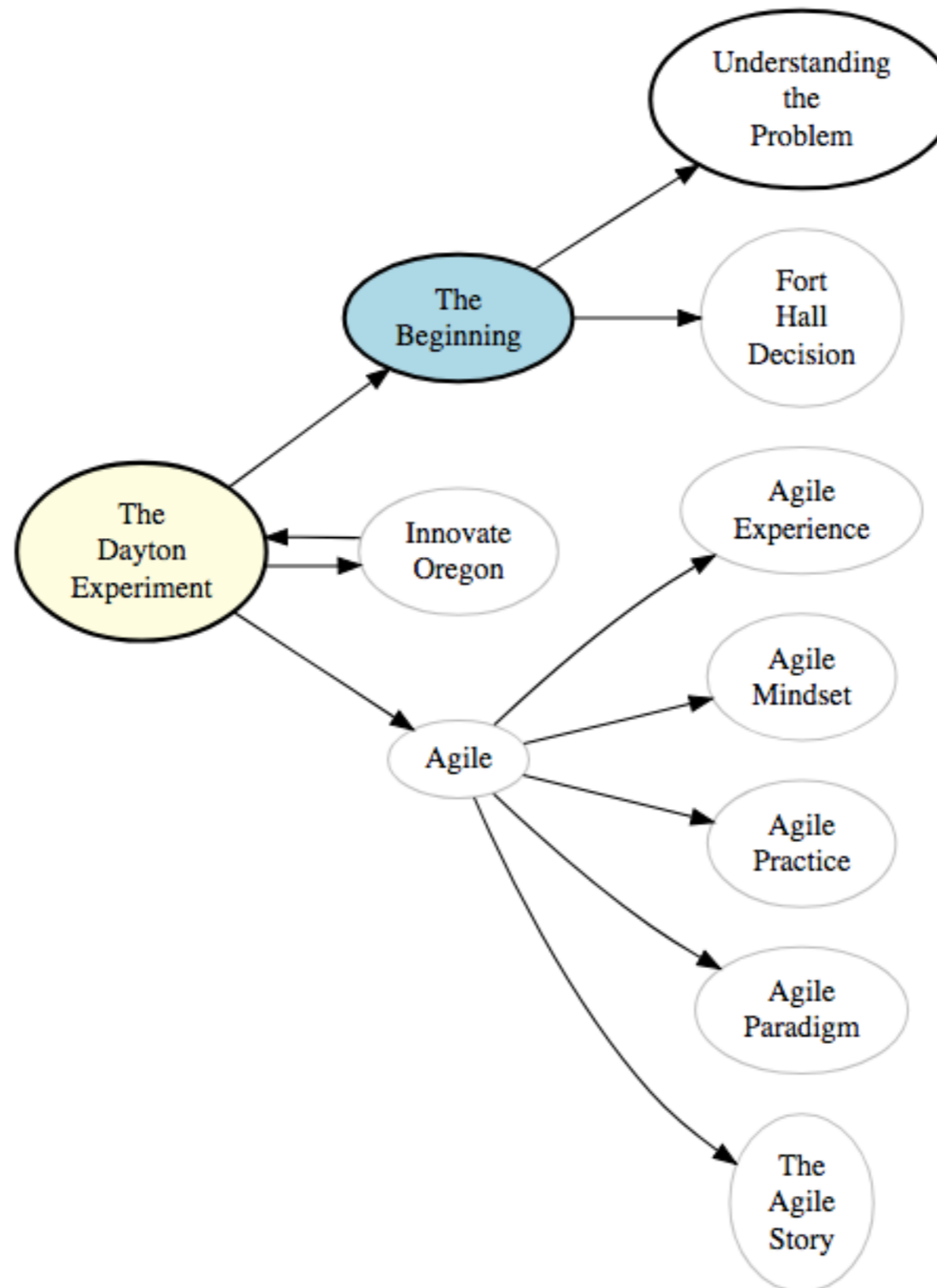
■ Graceful Extensibility

The opposite of brittleness in complex systems. David Woods presents a theory of graceful extensibility. This paper is dense and will reward further study. Know that it is not for the faint of heart. pdf ⬀

All systems have an envelope of performance, or a range of adaptive behavior, due to **finite resources** and **continuous change**. There is a transition zone where a system shifts regimes of performance when pushed to the edge of its envelope. Brittleness and graceful extensibility describe how the system responds while exceeding its envelope of performance. Graceful extensibility refers to a system's ability to adapt how it works to extend performance past the boundary into a new regime of performance invoking new resources, responses, relationships, and priorities.

At the heart of the theory of graceful extensibility is the fundamental concept of managing **risk of saturation** via regulating the **capacity for maneuver**.

XP and Normative Good

Extreme Programming (xp) prescribes a dozen practices that reinforce each other so as to allow teams to make software development decisions. We examine the distribution of responsibilities required for, and means to arrive at, good decisions within this framework.

Velocity provides a rough measure of this complex social process. Achieving and maintaining the process is more important for long term success than any goal one might set for velocity

Process

Programs consist of many thousands of statements about how a computer should usefully work in the future. We call individuals who make these statements programmers and those who decide what would be useful, customers.

Extreme programming exploits the modularity that became available in the '80s to allow programmers to make and

■ XP and Normative Good

Extreme Programming (xp) prescribes a dozen practices that reinforce each other so as to allow teams to make software development decisions. We examine the distribution of responsibilities required for, and means to arrive at, good decisions within this framework.

Process based on delivery.

Technique preserves options.
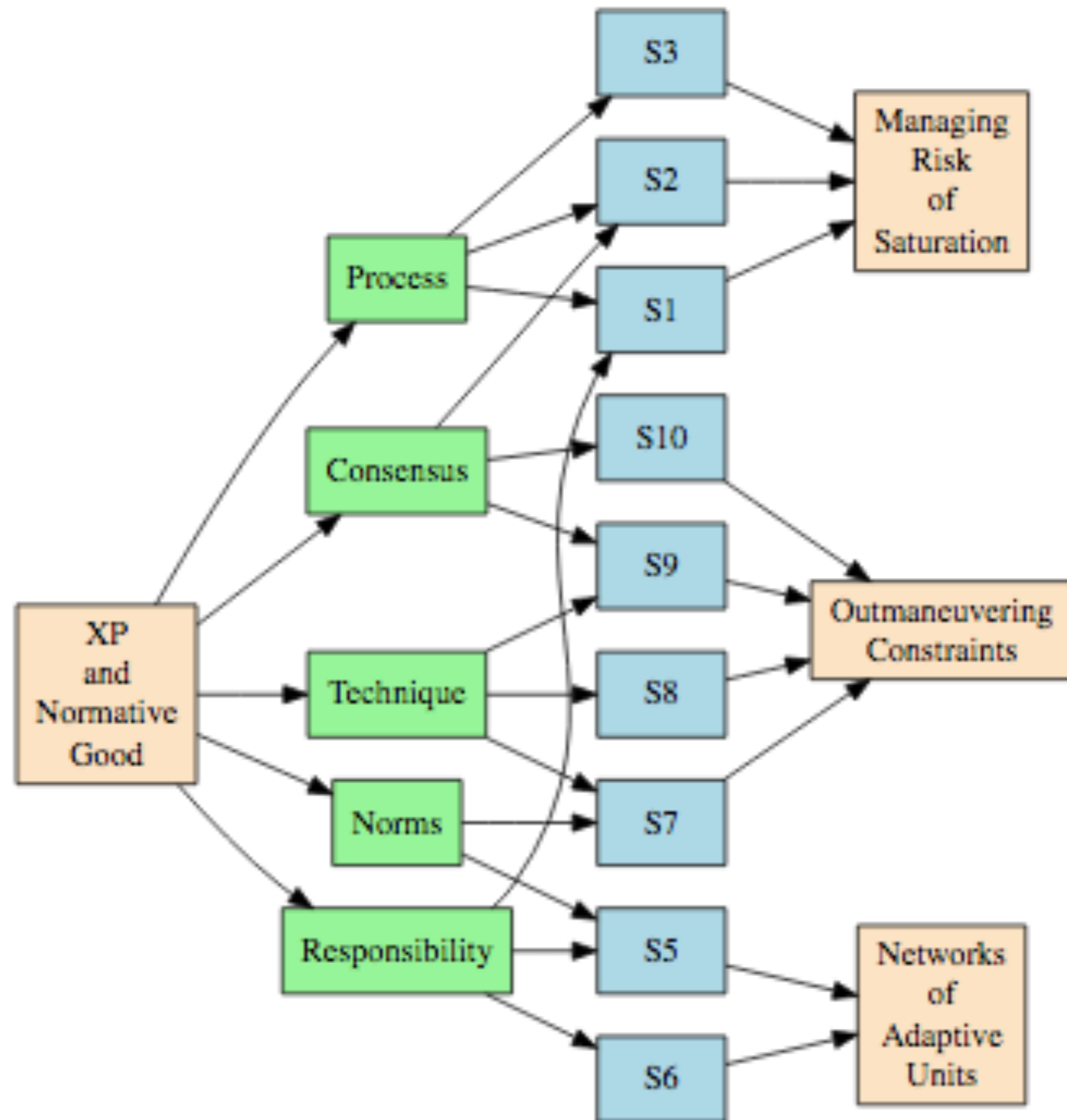
Responsibility over what one knows.

Consensus based on individual and shared experience.

Norms emerge from good decisions.

Velocity provides a rough measure of this complex social process. Achieving and maintaining the process is more important for long term success than any goal one might set for velocity

important for long term success than any goal one might set for velocity

Consensus

A team of developers can be effective for the long term to the degree that they can make the good decisions regarding code cleanliness that extreme programming reserves for them. S2

Individual experience varies widely even among skilled developers. Only shared experience in a particular codebase will lead to consensus. Reading each other's code helps but cannot duplicate that of programming together while discussing the present and future impact of every decision made. S9

Test-driven development enables pair-programming by sequencing new experience based more on recent experience than on deep personal experience unavailable to a companion. S10

older

S2

S2: Events will occur outside the bounds and will challenge the adaptive capacity of any unit, therefore, surprise continues to occur and demands response, otherwise the unit is brittle and subject to collapse in performance. Continuous Surprise

Subset A: Managing Risk of Saturation

Consensus

A team of developers can be effective for the long term to the degree that they can make the good decisions regarding code cleanliness that extreme programming reserves for them. S2

Individual experience varies widely even among skilled developers. Only shared experience in a particular codebase will lead to consensus. Reading each other's code helps but cannot duplicate that of programming together while discussing the present and future impact of every decision made. S9
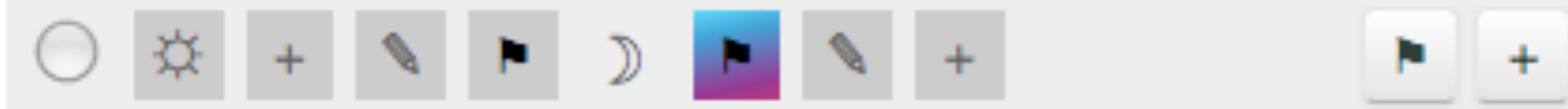
Test-driven development enables pair-programming by sequencing new experience based more on recent experience than on deep personal experience unavailable to a companion. S10

important for long term success than any goal one might set for velocity

important for long term success than any goal one might set for velocity

```
DOT strict digraph
  node [shape=box style=filled]
  node [fillcolor=bisque]
  rankdir=LR
  HERE NODE
    WHERE /^\[\[/
      node [fillcolor=palegreen]
      LINKS HERE -> NODE
        HERE
          WHERE /\[\[S\d+\]\]/
            node [fillcolor=lightblue]
            LINKS HERE -> NODE
              HERE
                WHERE /^Subset/
                  node [fillcolor=bisque]
                  LINKS HERE -> NODE
```

XP
and
Normative
Good

Consensus

S2

Managing
Risk
of
Saturation
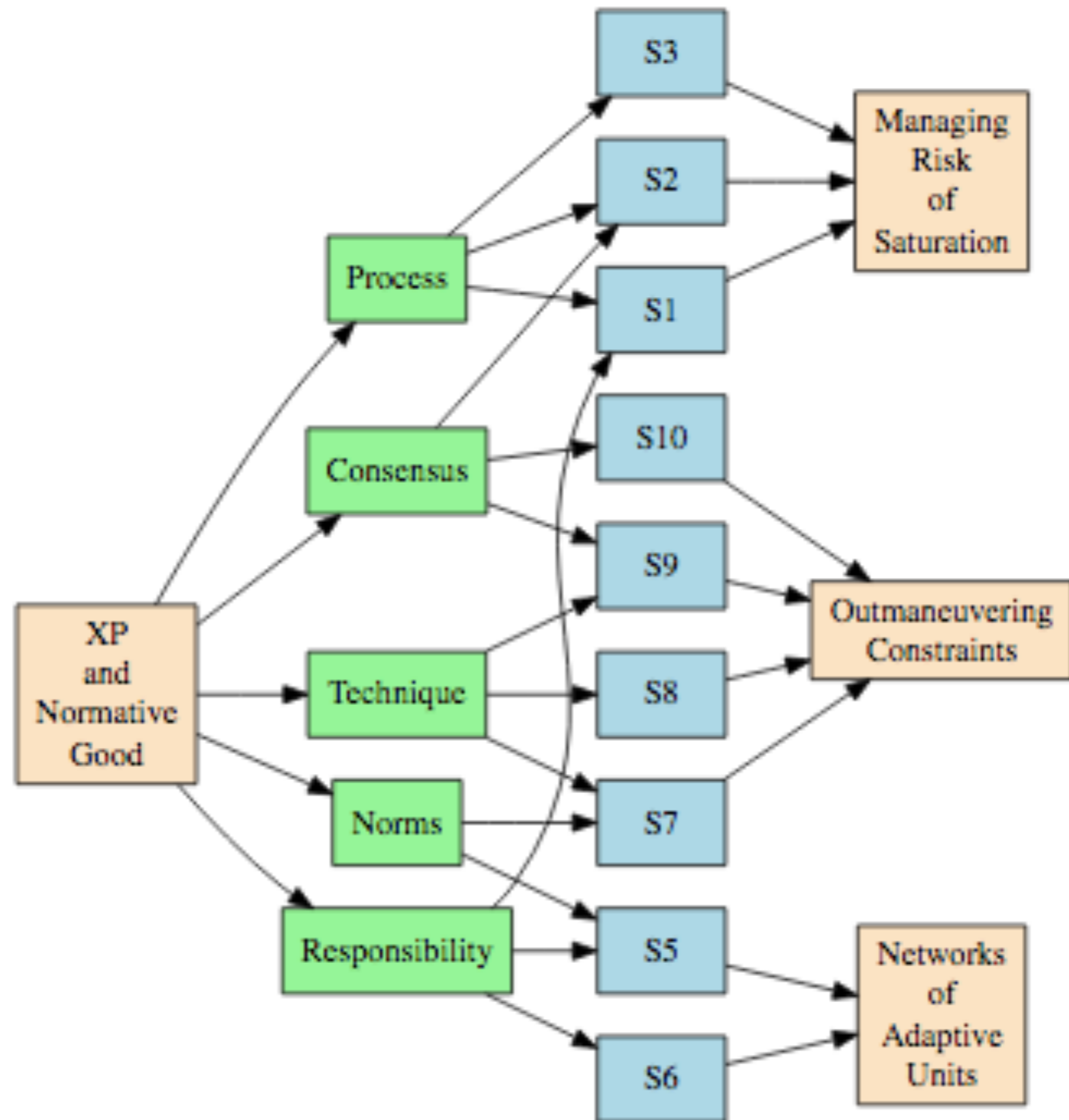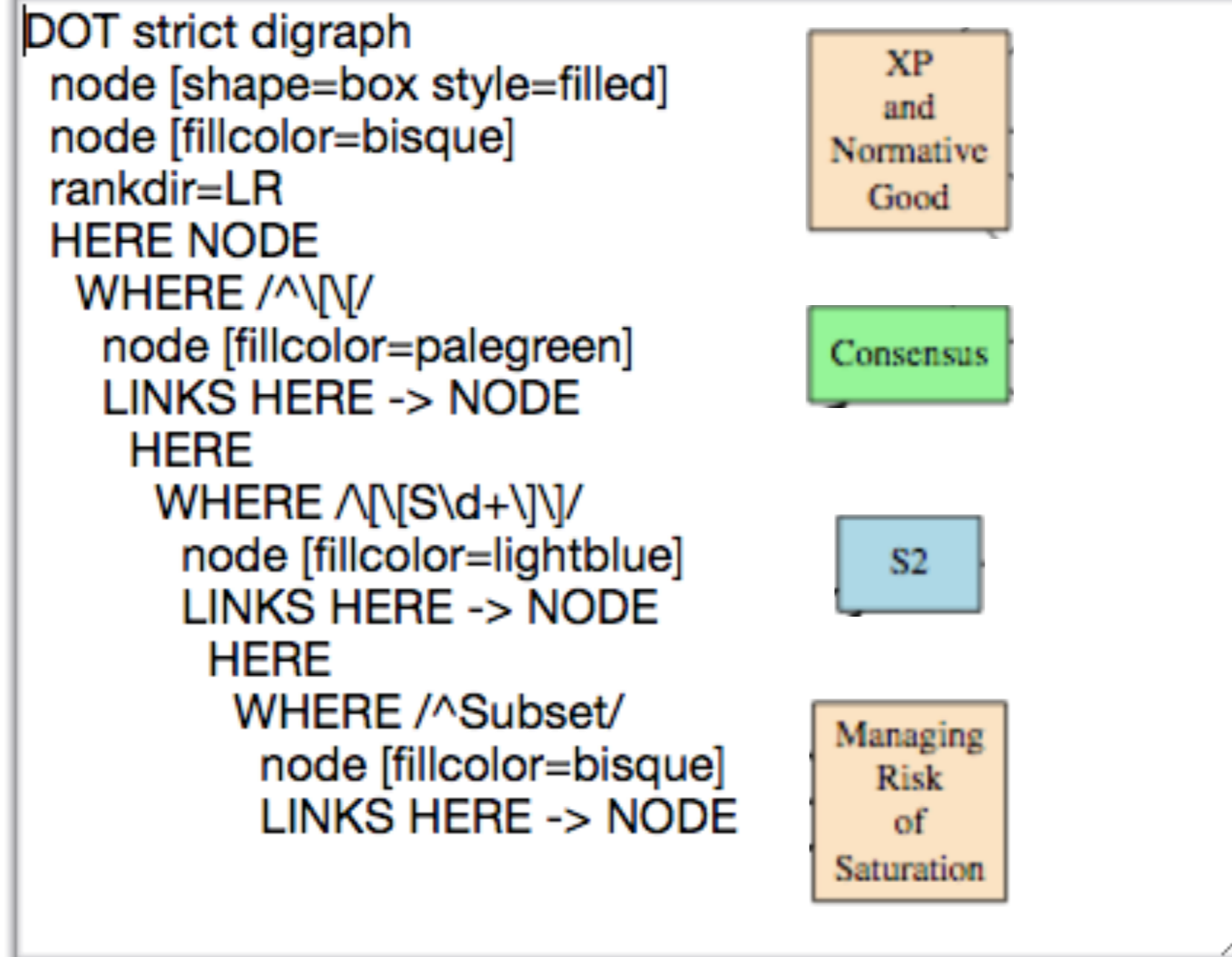
See Core Agile for a more concise description of the responsibility distribution described here.

See Infected with ROI and Marginal Distinction for alternative perspective on velocity.

important for long term success than any goal one might set for velocity

Summary

Diagram 1 (map):

Seattle 90 Ellensburg Butte 90
5 82 Pocatello
Portland Hermiston 84 86
San Francisco Sacramento 84 Salt Lake City
80 80
5 15

Diagram 2:

Common Areas At The Heart
Light On Two Sides Of Every Room
Window Place
Cascade Of Roofs
Number
Farmhouse Kitchen
Roof Layout

Diagram 3:

The Beginning
The Dayton Experiment
Innovate Oregon
Agile

Diagram 4:

TEAM
MEMBER
team store 1
LEADER 2
FORMER
DEPARTMENT CONTAINS
rpm accounts
6
PERSON
CONTAINS 5
workday 5
CONTAINS
DIVISION
12

Diagram 5:

Sanctuary
Special Use Patterns
400M Through Street Network
Market Center
School Campus
Public Space Patterns
Hospital
Place Network
Walkable Streetscape

Diagram 6:

XP and Normative Good
Consensus
Technique
Norms
Responsibility
S10
S9
S8
S7
S5
S6
Outmaneuvering Constraints
Networks of Adaptive Units