CSE 5544: Introduction to Data Visualization

Raghu Machiraju
machiraju.1@osu.edu

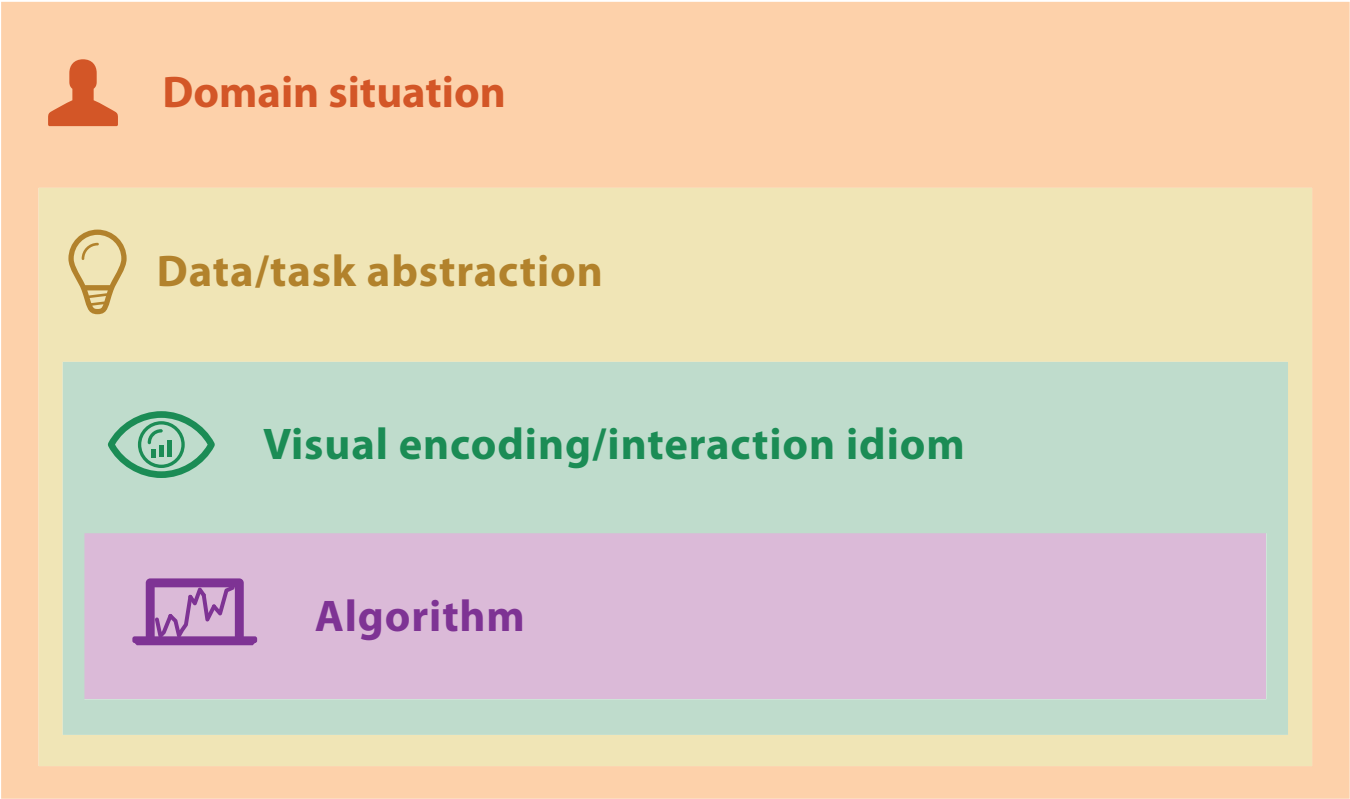
[xkcd]

Design study methodology

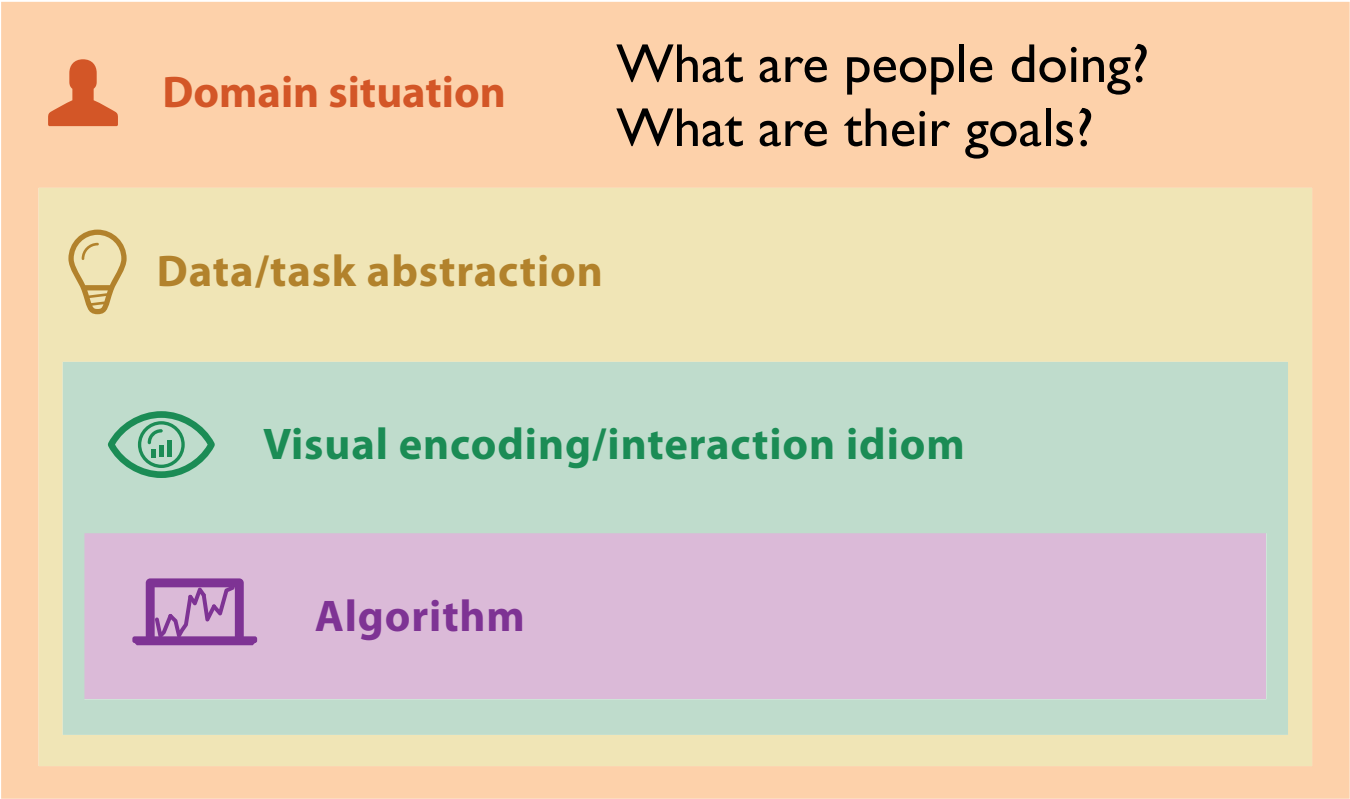
Industrial Engineer

Industrial engineering is a branch of engineering which deals with the optimization of complex processes or systems. Industrial engineers work to eliminate waste of time, money, materials, man-hours, machine time, energy and other resources that do not generate value.

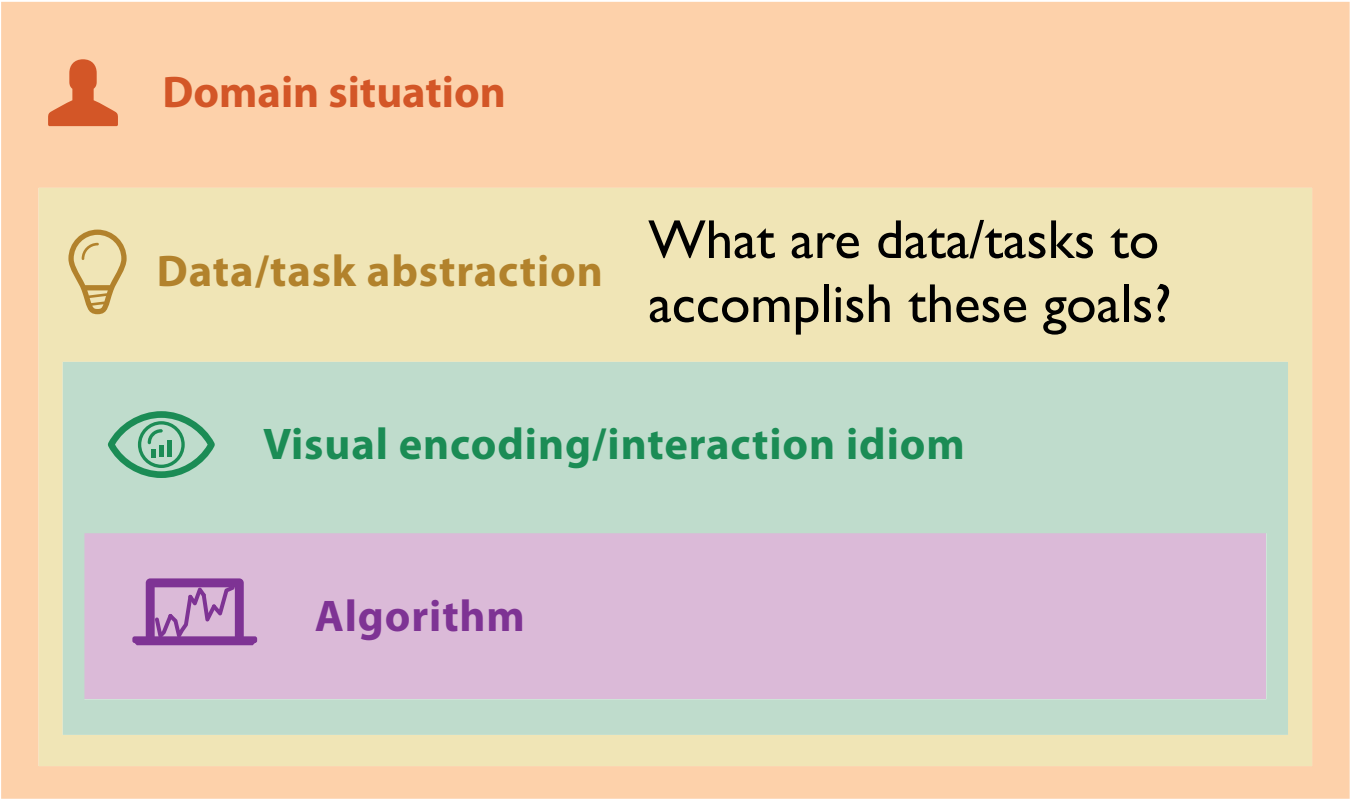
Nested model for vis design



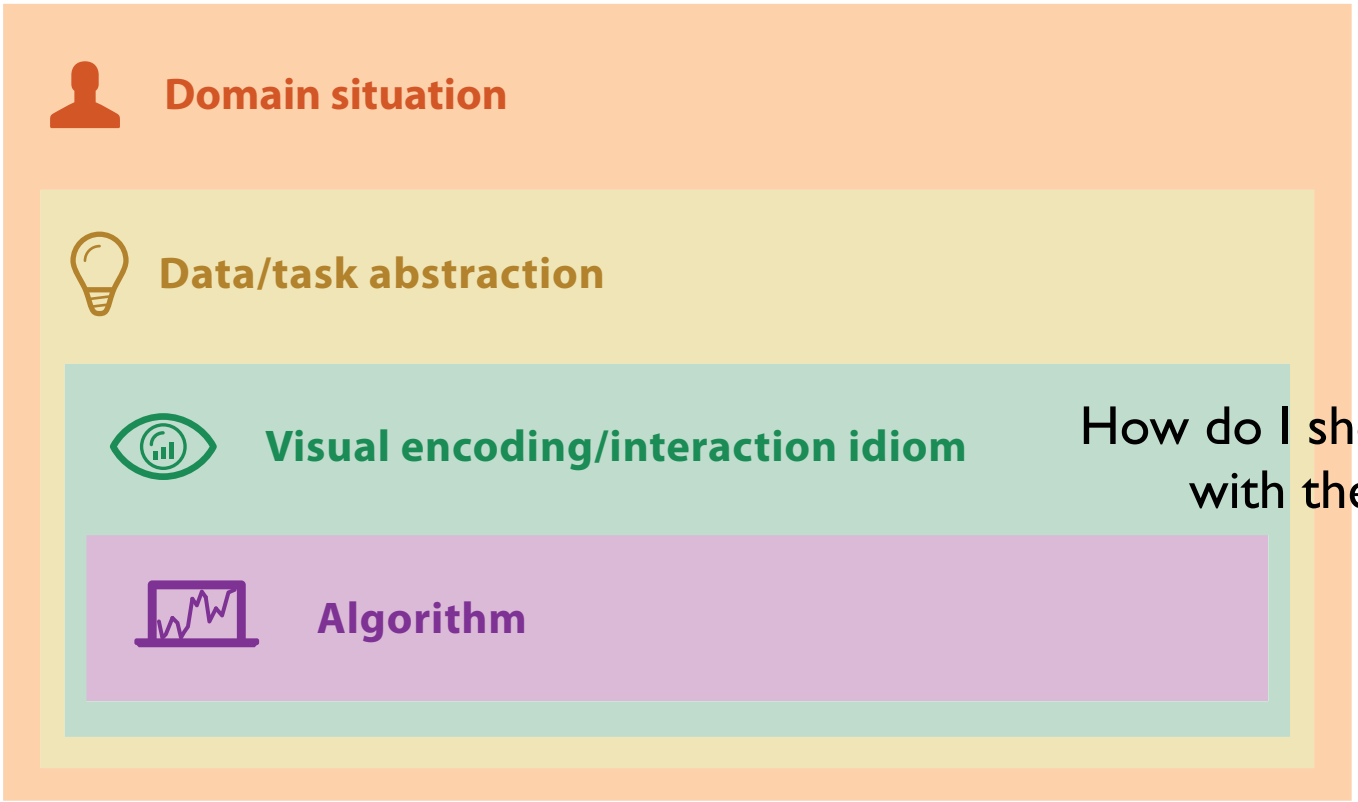
Nested model for vis design



Nested model for vis design

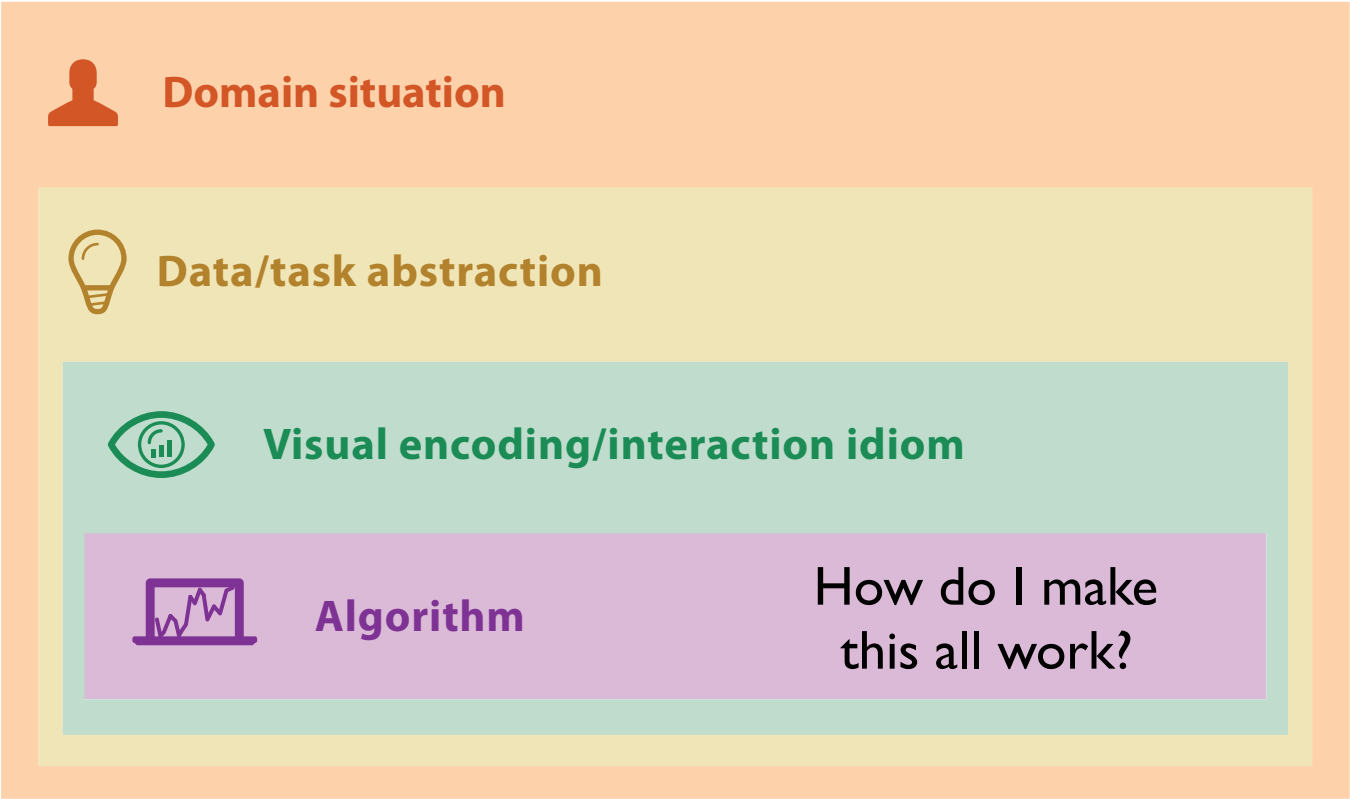


Nested model for vis design



How do I show/interact with the data?

Nested model for vis design



Nested model for vis design

Domain situation
You misunderstood their needs

Data/task abstraction
You're showing them the wrong thing

Visual encoding/interaction idiom
The way you show it doesn't work

Algorithm
Your code is too slow

Nested model for vis design

Domain situation
Observe target users using existing tools

Data/task abstraction

Visual encoding/interaction idiom
Justify design with respect to alternatives

Algorithm
Measure system time/memory
Analyze computational complexity

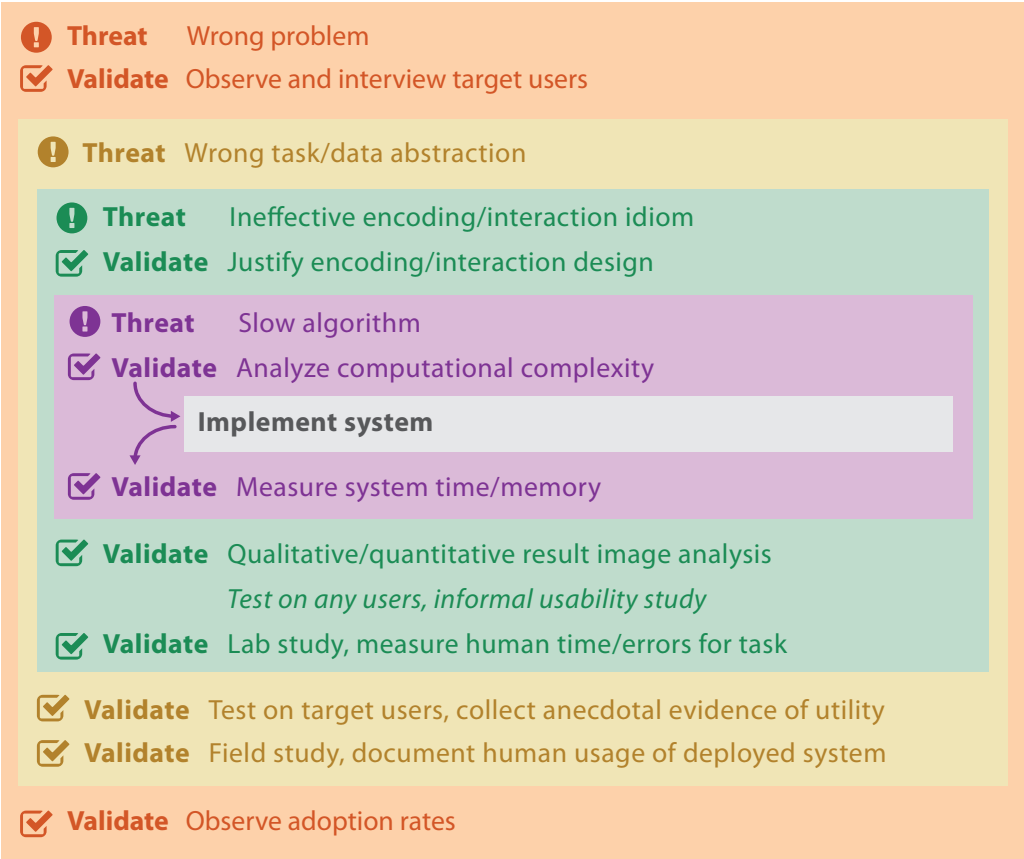
Analyze results qualitatively

Measure human time with lab experiment (*lab study*)

Observe target users after deployment (*field study*)

Measure adoption

Nested model for vis design



Workflow for designing a tool

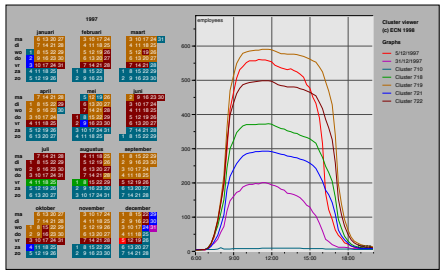
Making the right tool

Questions

Data

Tasks

Vis researcher



van Wijk:1999

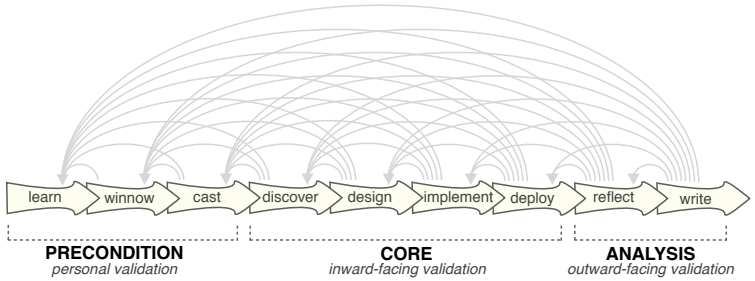
http://halalfocus.net/wp-content/uploads/2014/05/question-mark-nothing.jpg

Making the right tool

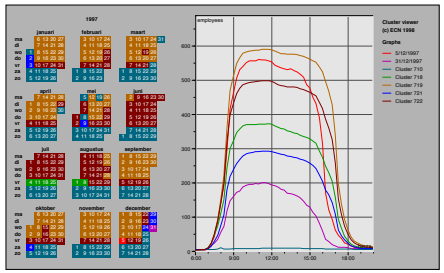
Questions

Data

Tasks

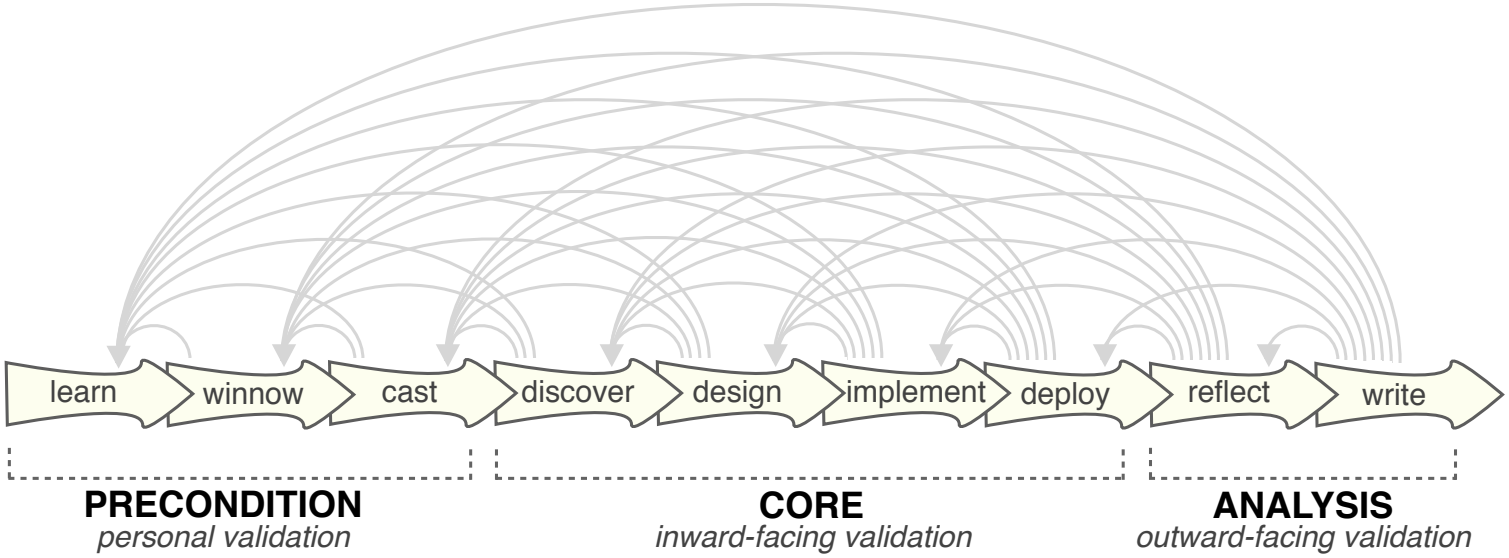


Design study methodology



van Wijk:1999

Design study methodology



Sedlmair:2012

Design study definition

Design study papers explore the choices made when applying infovis techniques in an application area, for example relating the visual encodings and interaction techniques to the requirements of the target task. Although a limited amount of application domain background information can be useful to provide a framing context in which to discuss the specifics of the target task, the primary focus of the case study must be the infovis content. Describing new techniques and algorithms developed to solve the target problem will strengthen a design study paper, but the requirements for novelty are less stringent than in a Technique paper.

[InfoVis03 CFP, infovis.org/infovis2003/CFP]

Munzner

Design study definition

Design study papers **explore the choices made when applying infovis techniques in an application area**, for example relating the visual encodings and interaction techniques to the requirements of the target task. Although a limited amount of application domain background information can be useful to provide a framing context in which to discuss the specifics of the target task, the primary focus of the case study must be the infovis content. Describing new techniques and algorithms developed to solve the target problem will strengthen a design study paper, but the requirements for novelty are less stringent than in a Technique paper.

[InfoVis03 CFP, infovis.org/infovis2003/CFP]

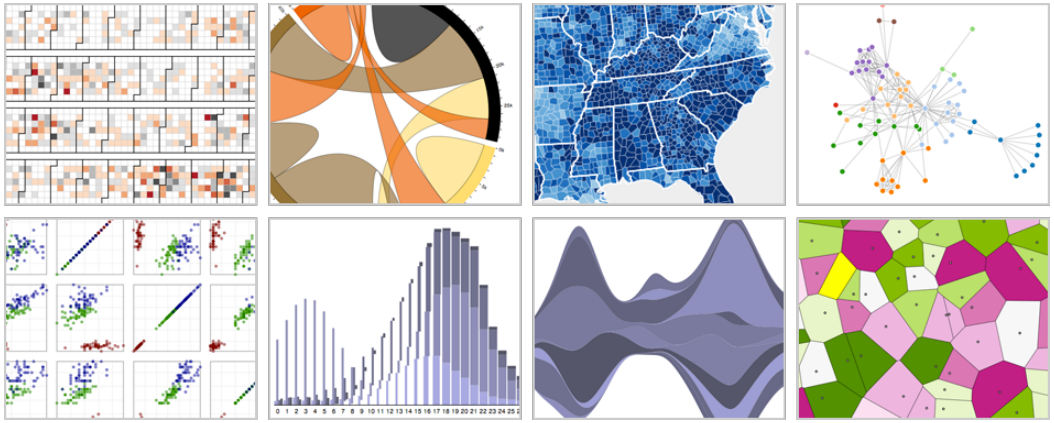
Munzner

Design study definition

Design study papers explore the choices made when applying infovis techniques in an application area, for example relating the visual encodings and interaction techniques to the requirements of the target task. Although a limited amount of application domain background information can be useful to provide a framing context in which to discuss the specifics of the target task, **the primary focus of the case study must be the infovis content**. Describing new techniques and algorithms developed to solve the target problem will strengthen a design study paper, but the requirements for novelty are less stringent than in a Technique paper.

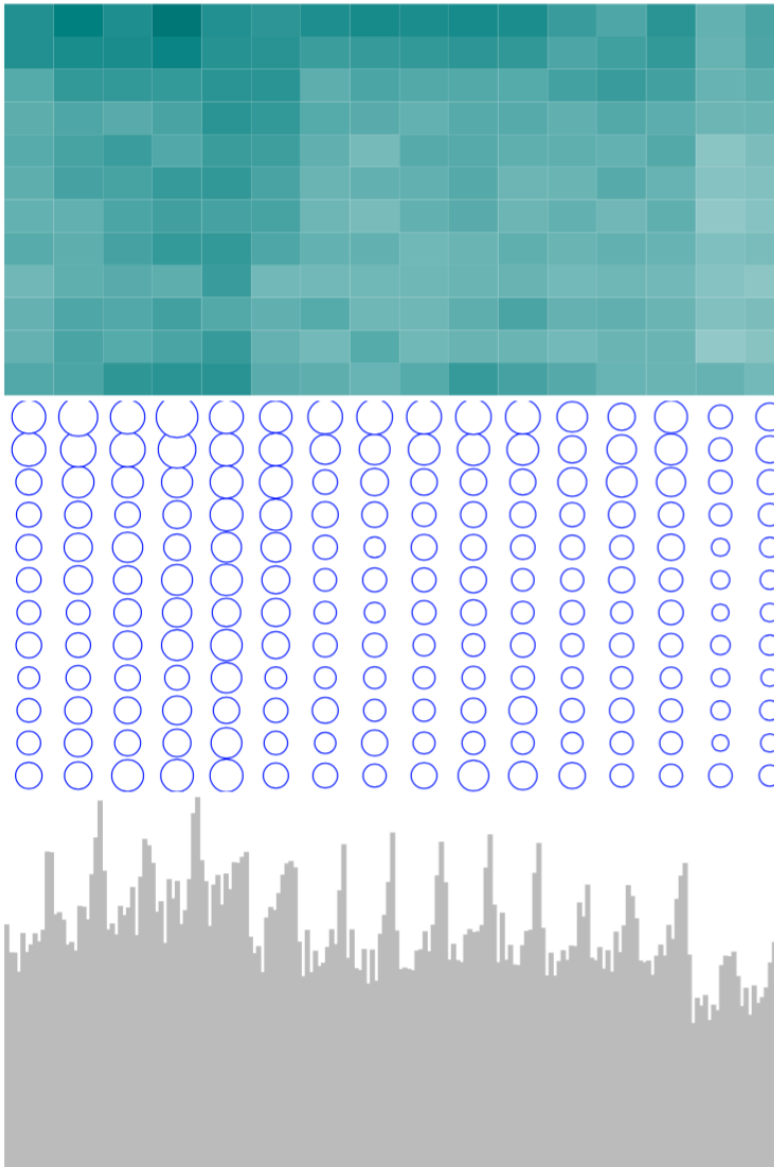
[InfoVis03 CFP, infovis.org/infovis2003/CFP]

Munzner



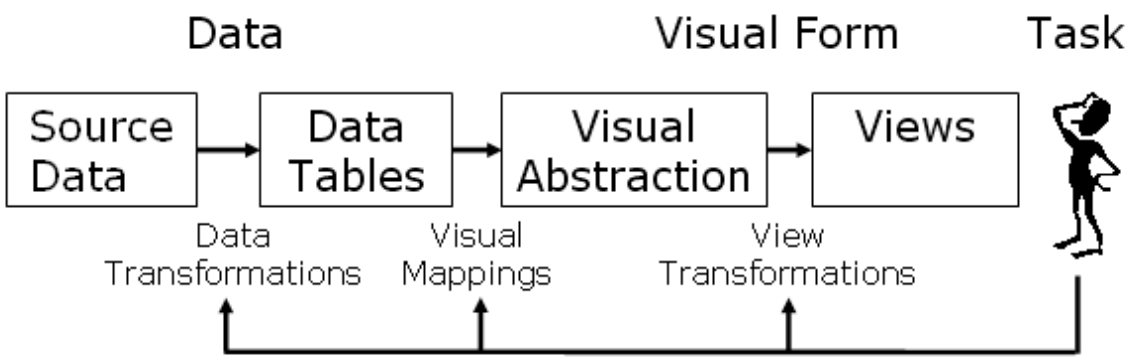
Very Useful

http://cscheid.net/courses/spr15/cs444/lectures/week3b.ht



d3's Pipeline Model

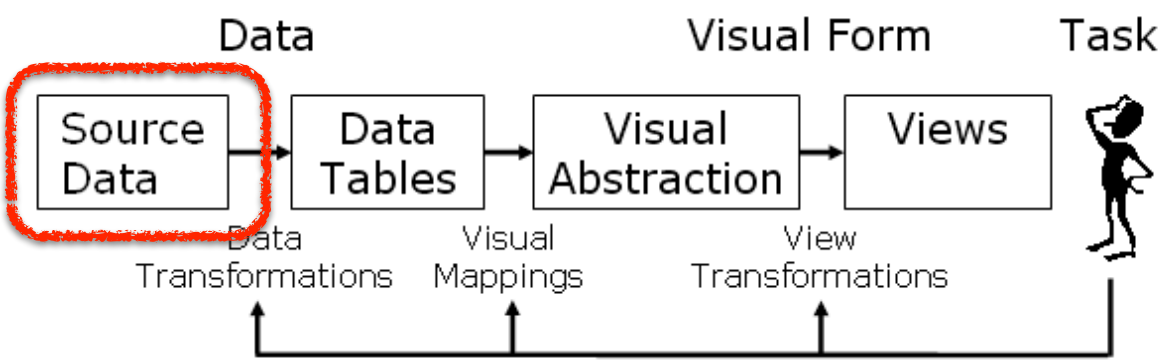
[J. Heer, Prefuse]



d3's Pipeline Model

[J. Heer, Prefuse]

```
d3.csv('http://www.example.com/data.csv', /* callback */)
d3.tsv('http://www.example.com/data.tsv', /* callback */)
d3.json('http://www.example.com/data.json', /* callback */)
```

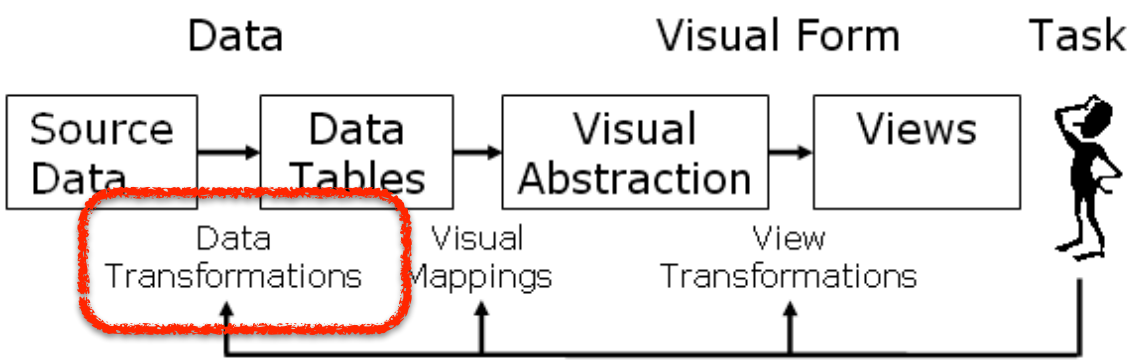


d3's Pipeline Model

[J. Heer, Prefuse]

```
d3.csv("f1.csv", function(error, data) {
  // Convert strings to numbers
  data.forEach(function(d) {
    d.year = +d.year;
    d.points = +d.points;
  });

  // Rest of vis code here
});
```

convert strings to numbers

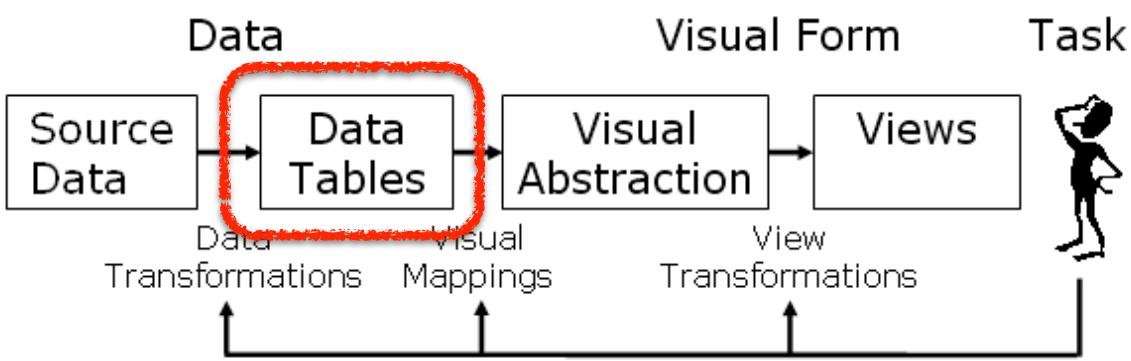


d3's Pipeline Model

[J. Heer, Prefuse]

```
d3.csv("f1.csv", function(error, data) {
  // Convert strings to numbers
  data.forEach(function(d) {
    d.year = +d.year;
    d.points = +d.points;
  });

  // Rest of vis code here
});
```

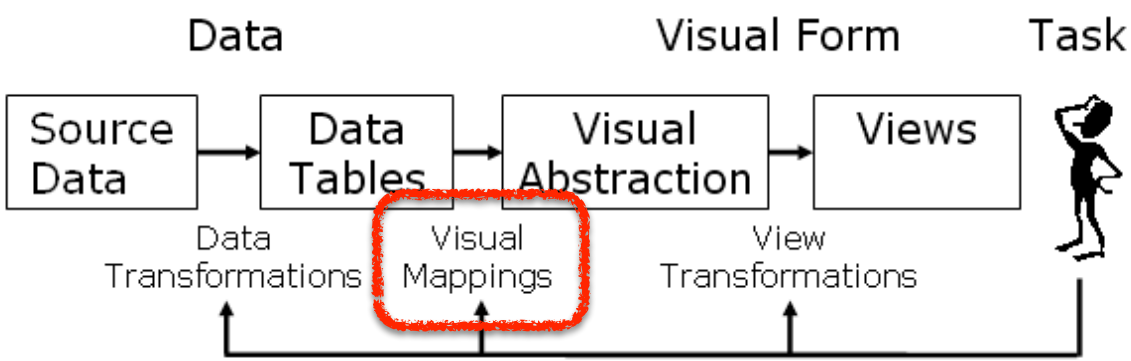
data are arrays



d3's Pipeline Model

[J. Heer, Prefuse]

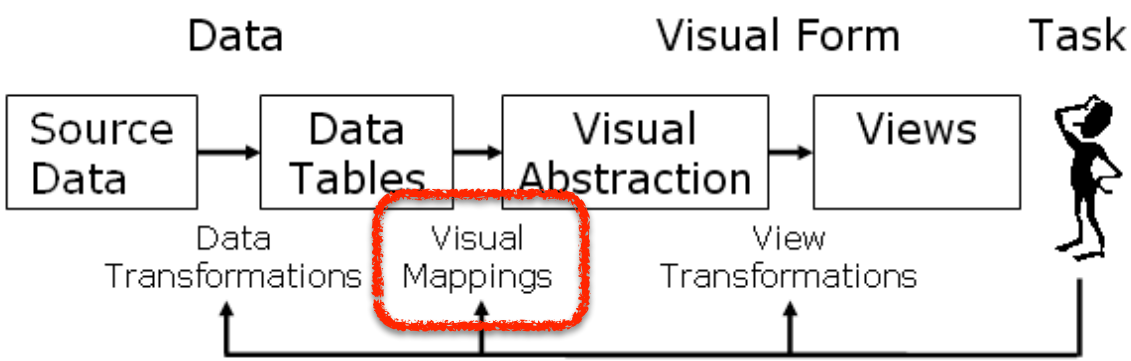
```
<circle class="dot" cx="599.9066147859922" cy="337.55364806866953" r="5"></circle>
<circle class="dot" cx="632.2645914396887" cy="372.9613733905579" r="5"></circle>
<circle class="dot" cx="588.2645914396887" cy="337.55364806866953" r="5"></circle>
```



d3's Pipeline Model

[J. Heer, Prefuse]

```
var circle = svg.selectAll("circle")
    .data(data);

circle.exit().remove();

circle.enter().append("circle")
    .attr("r", 5);

circle
    .attr("cx", function(d) { return d.x; })
    .attr("cy", function(d) { return d.y; });
```

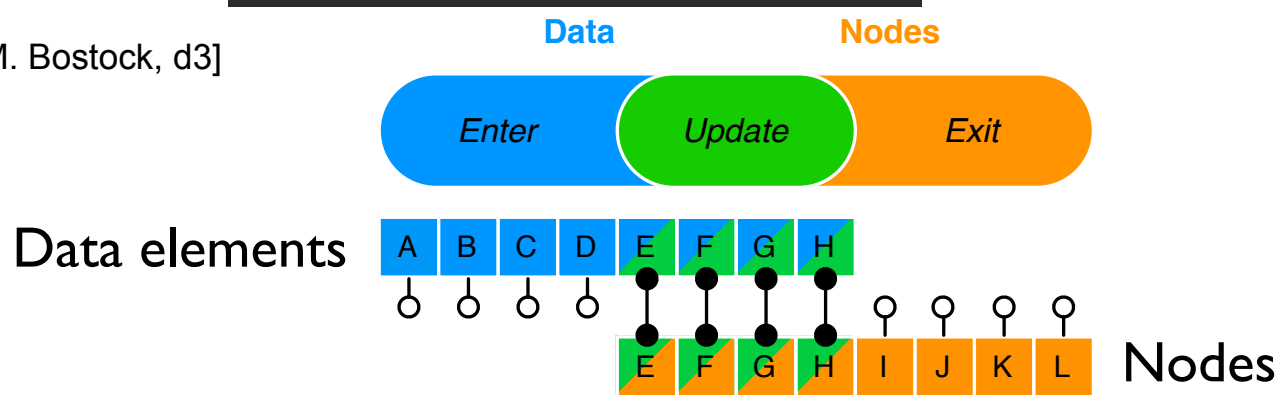


d3's Pipeline Model

```
var circle = svg.selectAll("circle")
    .data(data);

circle.exit().remove();

circle.enter().append("circle")
    .attr("r", 5);

circle
    .attr("cx", function(d) { return d.x; })
    .attr("cy", function(d) { return d.y; });
```

join with data

[M. Bostock, d3]

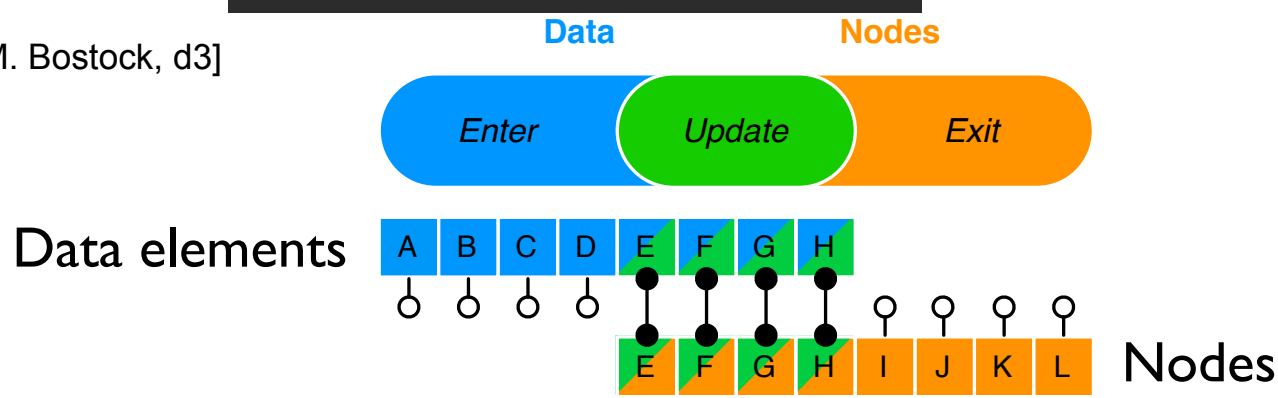


d3's Pipeline Model

```
var circle = svg.selectAll("circle")
    .data(data);

circle.exit().remove();

circle.enter().append("circle")
    .attr("r", 5);

circle
    .attr("cx", function(d) { return d.x; })
    .attr("cy", function(d) { return d.y; });
```

"exit" for nodes without data

[M. Bostock, d3]

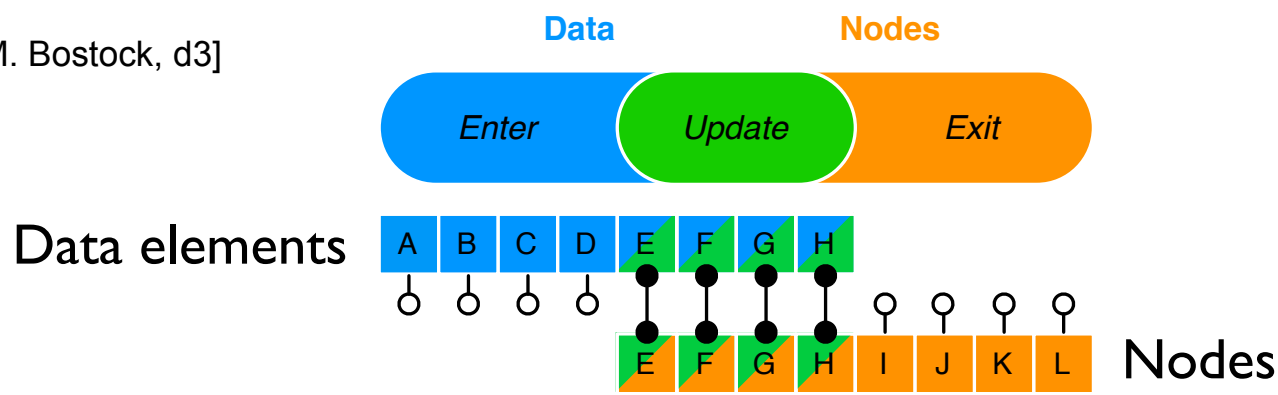


d3's Pipeline Model

```
var circle = svg.selectAll("circle")
    .data(data);

circle.exit().remove();

circle.enter().append("circle")
    .attr("r", 5);

circle
    .attr("cx", function(d) { return d.x; })
    .attr("cy", function(d) { return d.y; });
```

"enter" for data without nodes

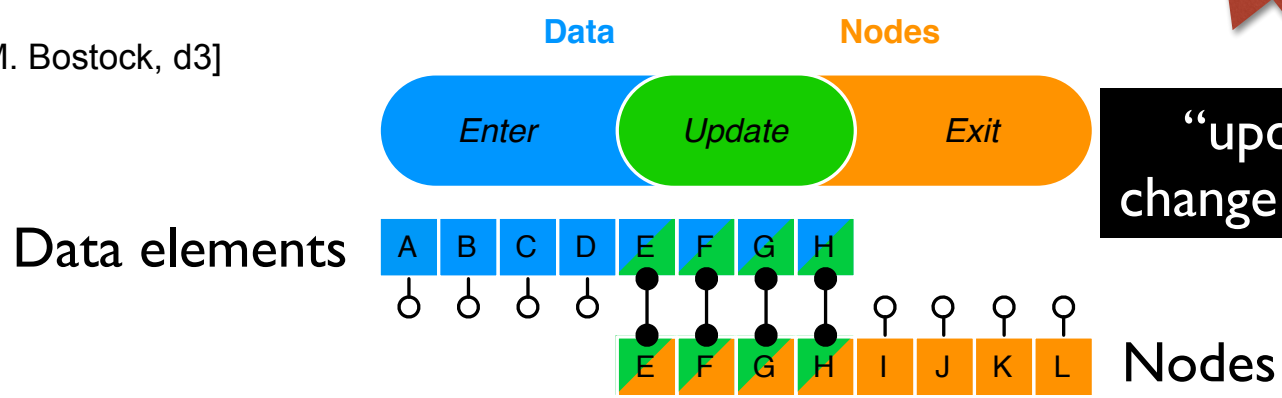
[M. Bostock, d3]



d3's Pipeline Model

```
var circle = svg.selectAll("circle")
    .data(data);

circle.exit().remove();

circle.enter().append("circle")
    .attr("r", 5);

circle
    .attr("cx", function(d) { return d.x; })
    .attr("cy", function(d) { return d.y; });
```

[M. Bostock, d3]

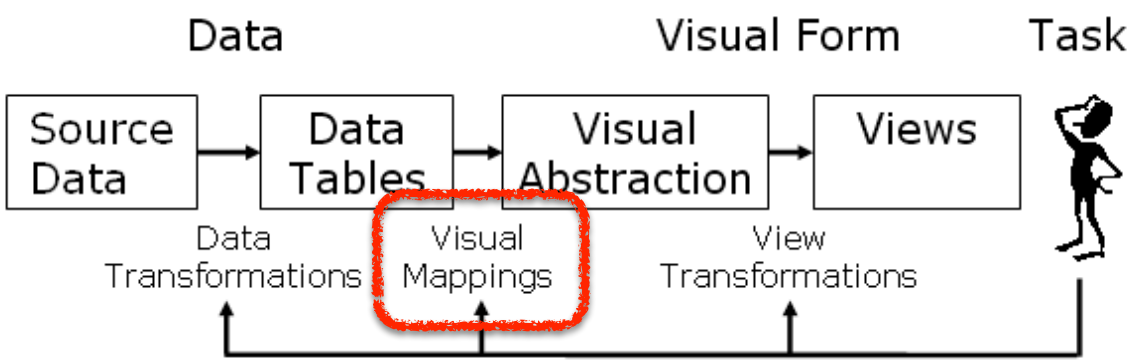


"update" to change attributes

d3's Pipeline Model

[J. Heer, Prefuse]

```
var y = d3.scale.linear()
    .range([height, 0])
    .domain([minPoints, maxPoints]);
```



d3's Pipeline Model

[J. Heer, Prefuse]

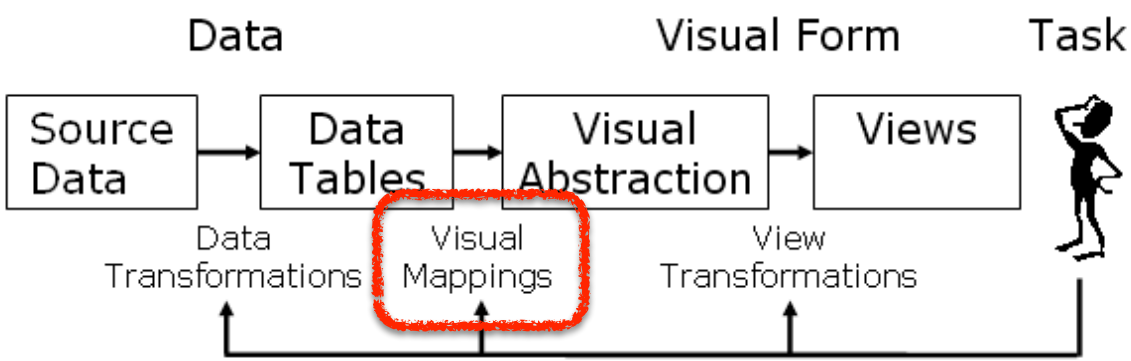
linear

log

ordinal

time

color

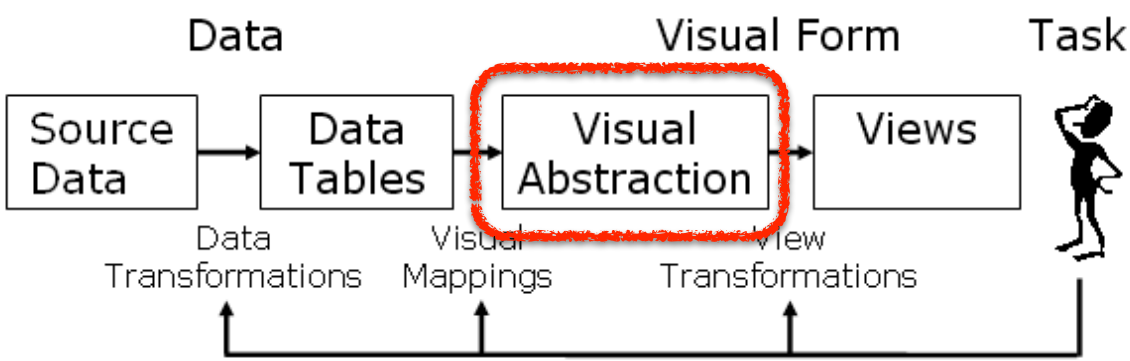


d3's Pipeline Model

[J. Heer, Prefuse]

uses svg/html for visual elements

```
<svg>
  <rect/>    <!-- rectangles -->
  <circle/>  <!-- circles -->
  <text/>    <!-- text -->
  <ellipse/> <!-- ellipses -->
</svg>
<html>
  <p/>       <!-- paragraph text -->
  <div/>
</html>
```

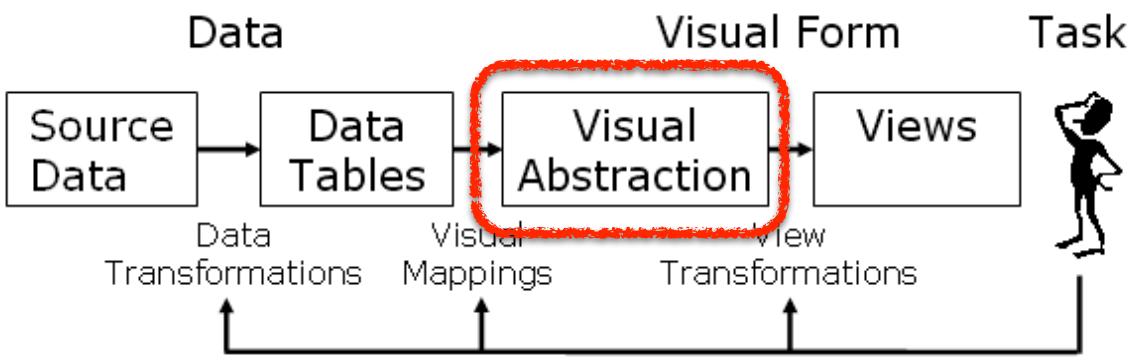


d3's Pipeline Model

[J. Heer, Prefuse]

functions for axes, layouts, paths

```
var xAxis = d3.svg.axis()
      .scale(mainX)
      .orient("bottom");
chart.append("g")
       .attr("class", "x axis")
       .attr("transform", "translate(0," + height + ")")
       .call(xAxis);
```



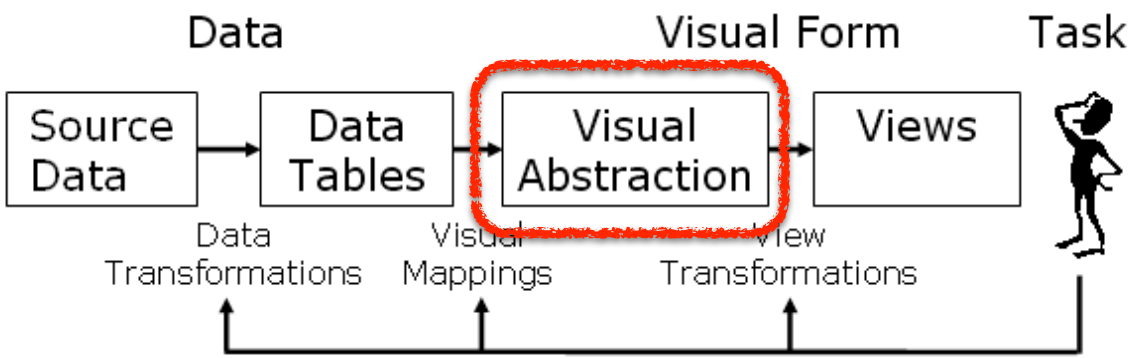
d3's Pipeline Model

[J. Heer, Prefuse]

functions for axes, layouts, paths

```
var xAxis = d3.svg.axis()
      .scale(mainX)
      .orient("bottom");
chart.append("g")
       .attr("class", "x axis")
       .attr("transform", "translate(0," + height + ")")
       .call(xAxis);
```

xAxis is a function!



d3's Pipeline Model

[J. Heer, Prefuse]

functions for axes, layouts, paths

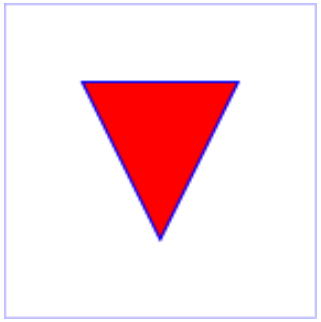
```
var layout = d3.layout.force()
    .linkDistance(50)
    .charge(-150)
    .size([width, height])
    .nodes(graph.nodes)
    .links(graph.links)
    .start();

// This gets updated every time the layout iterates
layout.on("tick", function() {
  node.attr("cx", function(d) {return d.x;})
      .attr("cy", function(d) {return d.y;});
  link.attr("x1", function(d) {return d.source.x;})
      .attr("y1", function(d) {return d.source.y;})
      .attr("x2", function(d) {return d.target.x;})
      .attr("y2", function(d) {return d.target.y;});
});
```

d3's Pipeline Model

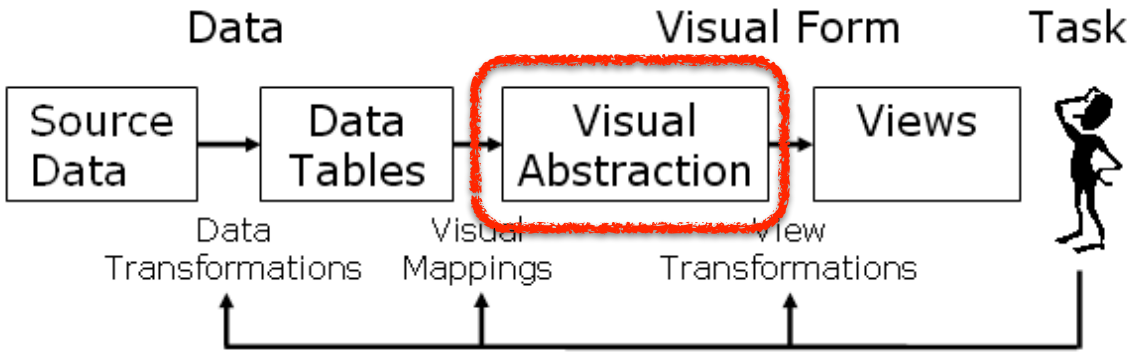
[J. Heer, Prefuse]

functions for axes, layouts, paths

```
<path d="M 100 100 L 300 100 L 200 300 z"
      fill="red" stroke="blue" stroke-width="3" />
```



http://www.w3.org/TR/SVG/paths.html#PathElement



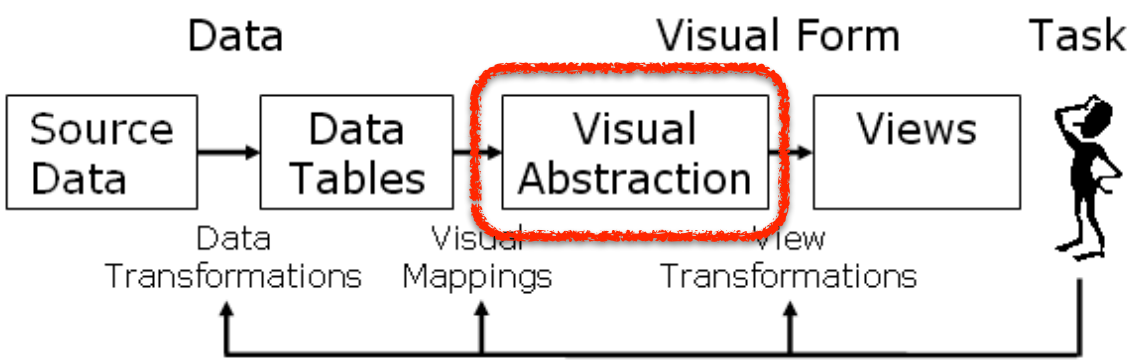
d3's Pipeline Model

[J. Heer, Prefuse]

functions for axes, layouts, paths

```
var lineFunction = d3.svg.line()
                    .x(function(d) { return d.x; })
                    .y(function(d) { return d.y; })
                    .interpolate("linear");
var lineGraph = svgContainer.append("path")
                        .attr("d", lineFunction(lineData))
                        .attr("stroke", "blue")
                        .attr("stroke-width", 2)
                        .attr("fill", "none");
```

https://www.dashingd3js.com/svg-paths-and-d3js



d3's Pipeline Model

[J. Heer, Prefuse]

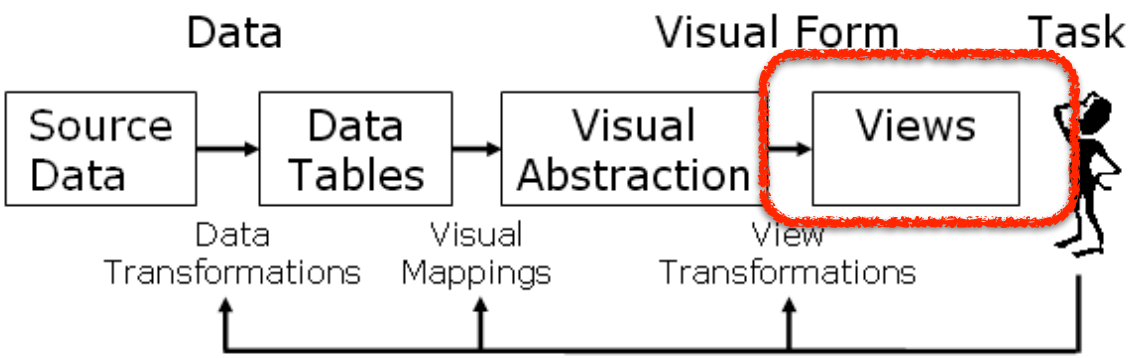
rendering done by browser engine

hardware-accelerated transitions

vis is part of the DOM

- many items can cause issues
- can view all vis elements

```
chart.selectAll(".group")
        .data(data)
        .transition().delay(100)
        .attr("transform", function(d) {
          return "translate("+mainX(d[mainKey])+",0)";
        }).selectAll("rect")
          .attr("width", subX.rangeBand())
          .attr("x", function(d) {return subX(d[subKey]);})
          .style("fill", function(d) {return colors(d[subKey]);});
```

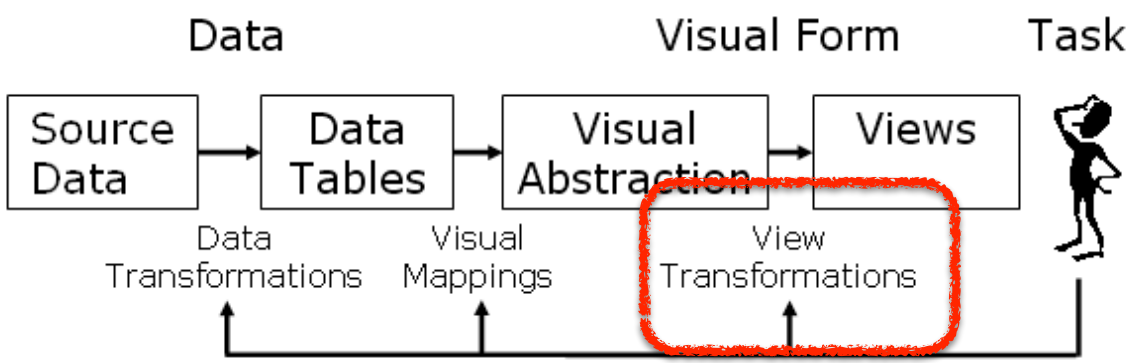


d3's Pipeline Model

[J. Heer, Prefuse]

```
chart.selectAll(".group")
        .data(data)
        .transition().delay(100)
        .attr("transform", function(d) {
          return "translate("+mainX(d[mainKey])+",0)";
        }).selectAll("rect")
          .attr("width", subX.rangeBand())
          .attr("x", function(d) {return subX(d[subKey]);})
          .style("fill", function(d) {return colors(d[subKey]);});
```

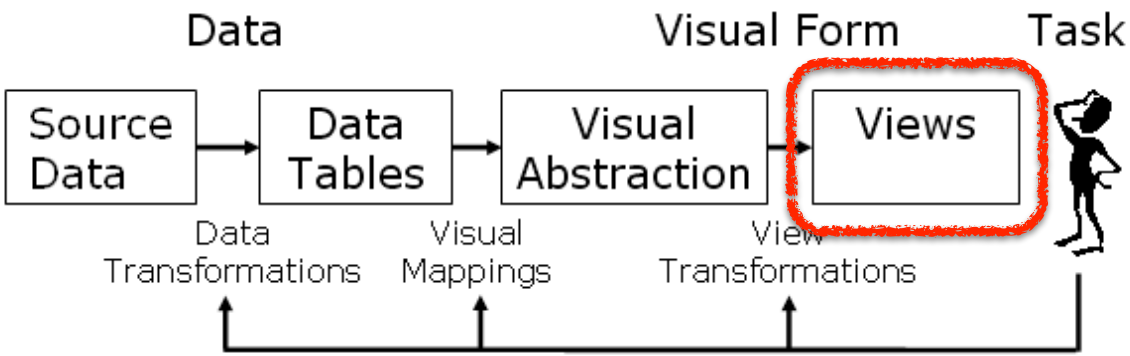
everything from here is animated



d3's Pipeline Model

[J. Heer, Prefuse]

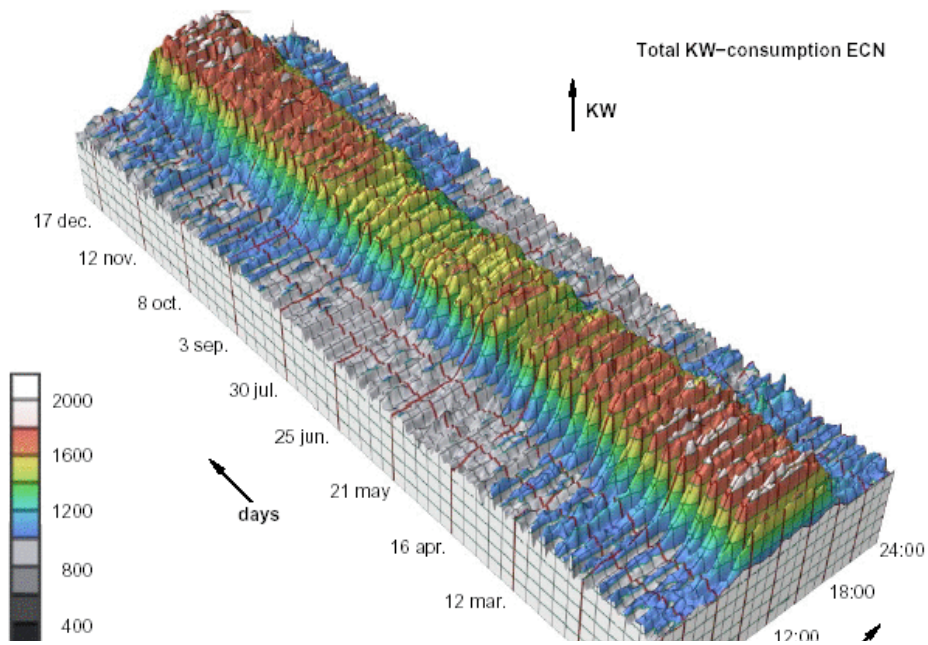
- rendering done by browser engine
- hardware-accelerated transitions
- vis is part of the DOM
  - many items can cause issues
  - can view all vis elements



Design Example

Time-series data: Case A naive

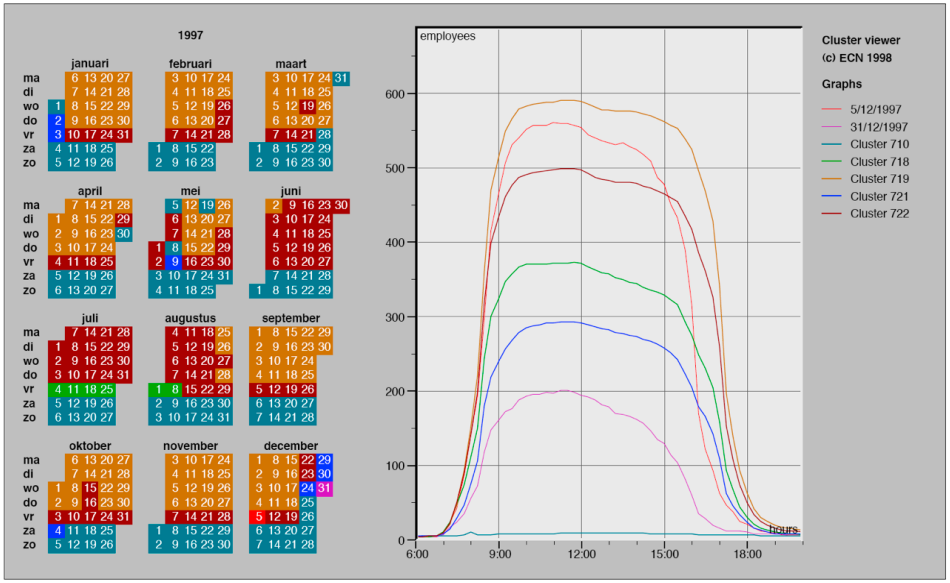
extruded curves: detailed comparisons impossible



[Cluster and Calendar based Visualization of Time Series Data. van Wijk and van Selow, Proc. InfoVis 99.]

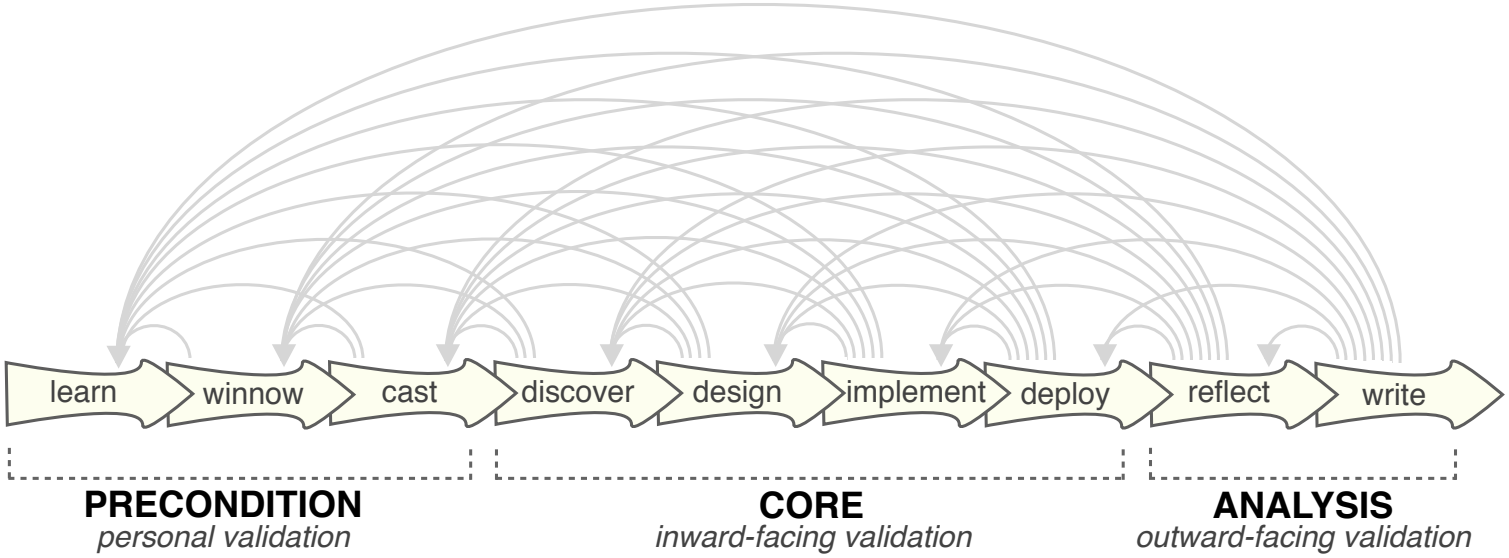
Case A: Better Cluster-Calendar Solution

- derived data: cluster hierarchy
- juxtapose multiple views: calendar, superimposed 2D curves



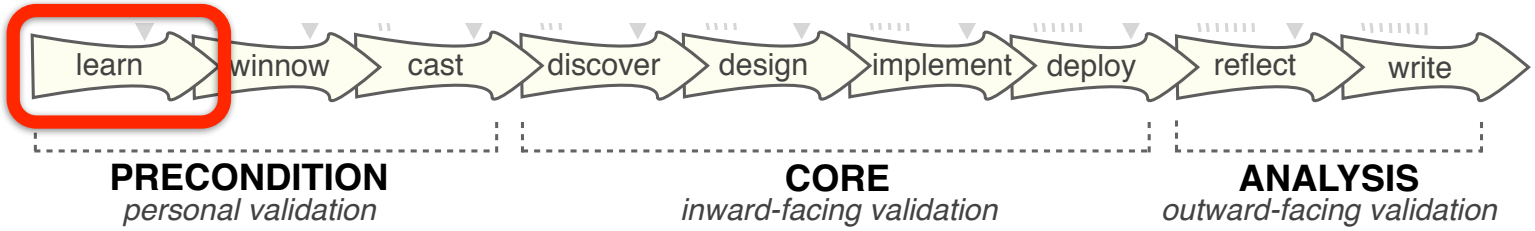
[Cluster and Calendar based Visualization of Time Series Data. van Wijk and van Selow, Proc. InfoVis 99.]

Design study methodology



Sedlmair:2012

Design study methodology



What tools/techniques are available?

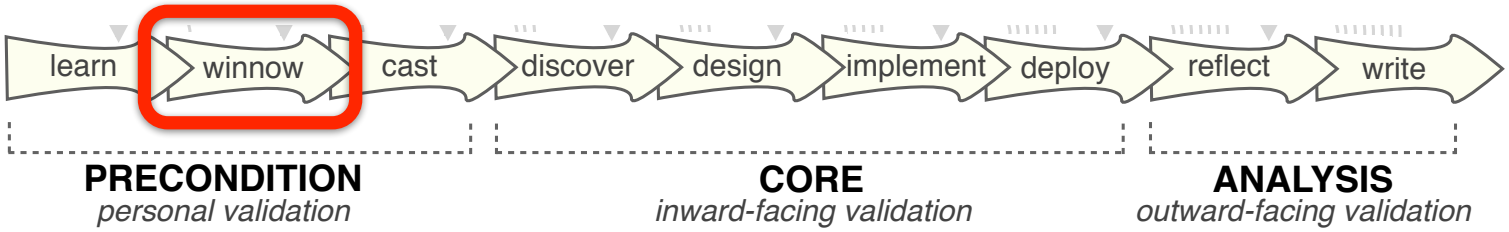
Read vis papers

Read vis books

Talk to vis practitioners

This course!

Design study methodology



Are these good collaborators?

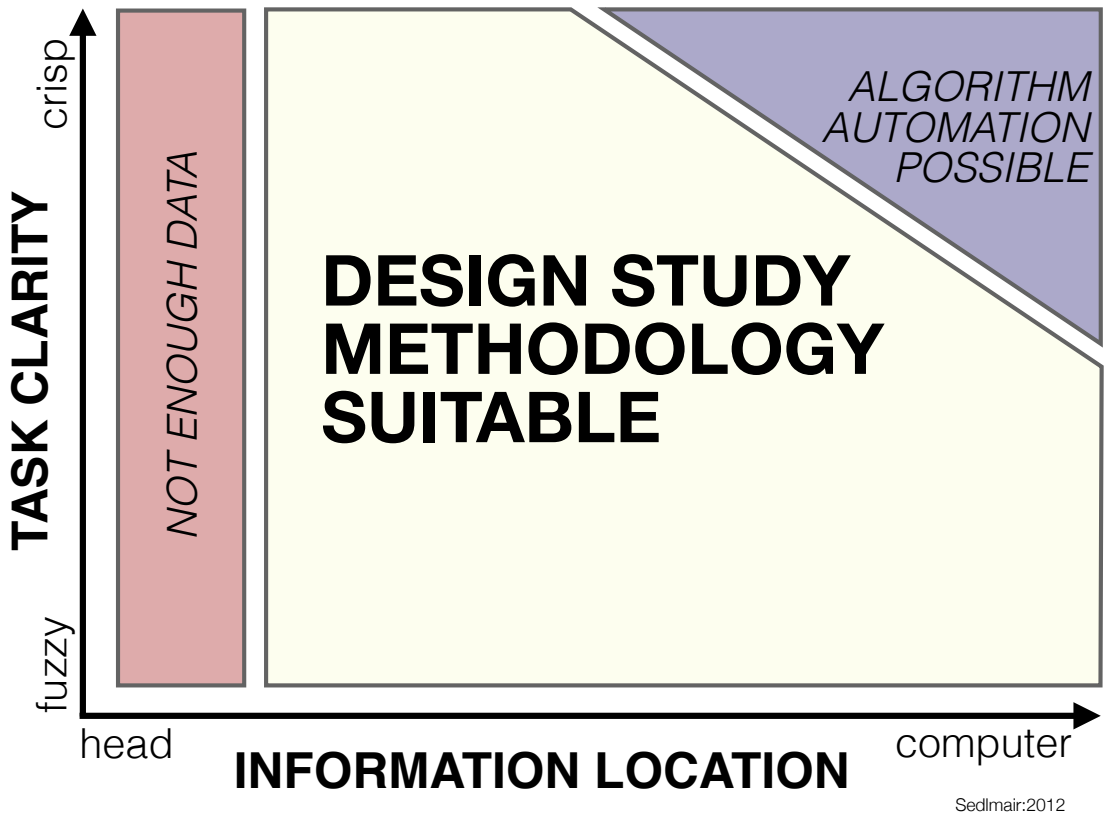
Do they have interesting problems?

Do they need novel solutions?

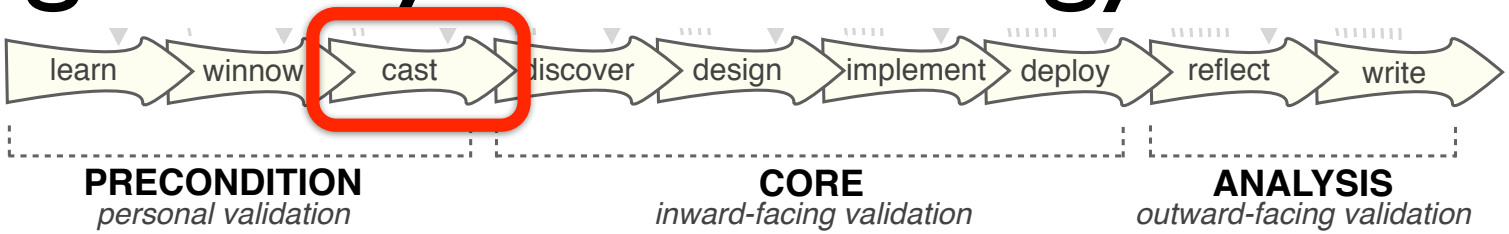
Is there data?

Can I work with these people?

When can you do a design study?



Design study methodology



Who's who?

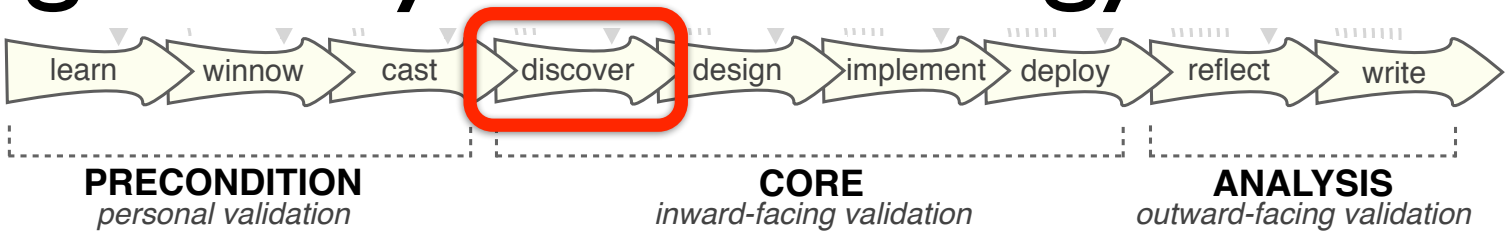
Do people have time for a new project?

"Front-line analyst" is the domain expert

Are there false "front-line analysts"?

Do you need a "translator"?

Design study methodology



Problem characterization and abstraction

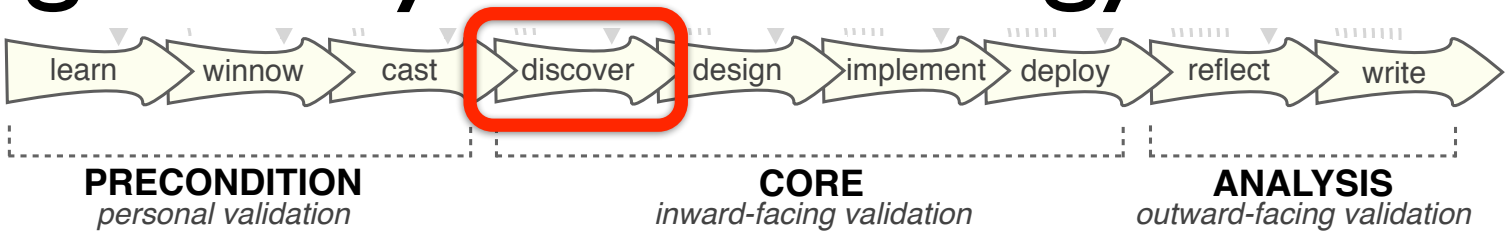
Requirements analysis

Critical reflection on requirements!

Abstraction is important for transferability

Need some domain-expert knowledge

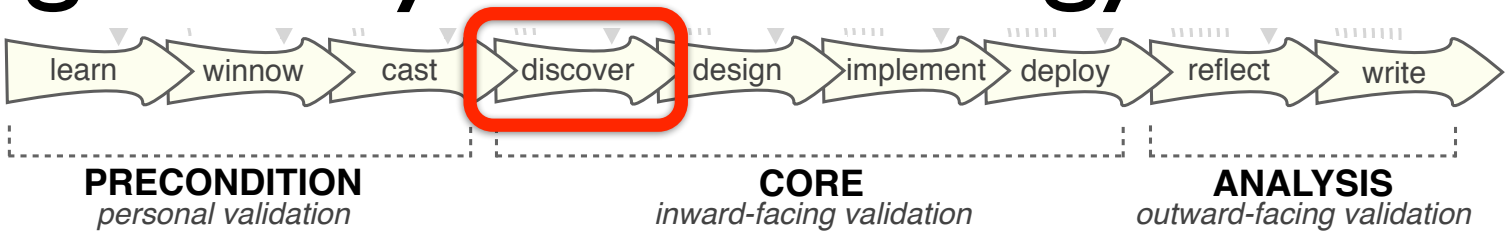
Design study methodology



example: Cluster-Calendar, van Wijk and van Selow

Overall goal: are there temporal patterns in power consumption?

Design study methodology



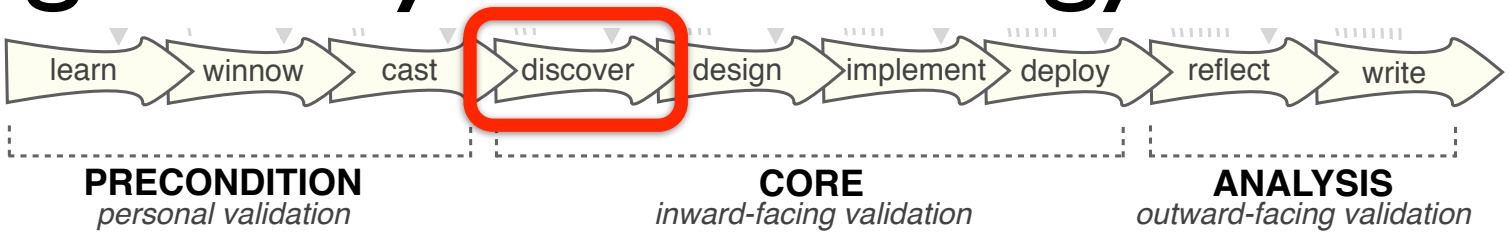
example: Cluster-Calendar, van Wijk and van Selow

Data: ~50K pairs of (value, time)

Tasks

- Find standard day patterns
- Find out how patterns are distributed over year, week, season
- Find outliers from standard daily patterns
- Want overview first, details on demand

Design study methodology



example: Cluster-Calendar, van Wijk and van Selow

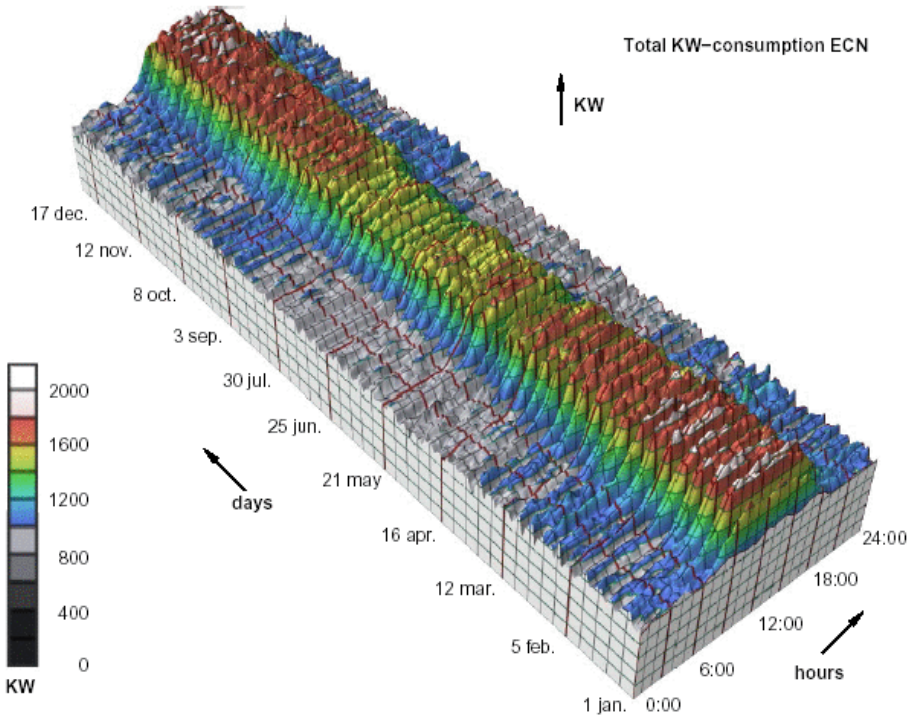
Limitations of previous work:

predictive mathematical models: details lost

scale-space approaches (wavelet, fourier, fractal): hard to interpret, known scales lost

3D mountain diagram (x: hours, y: value, z: days)

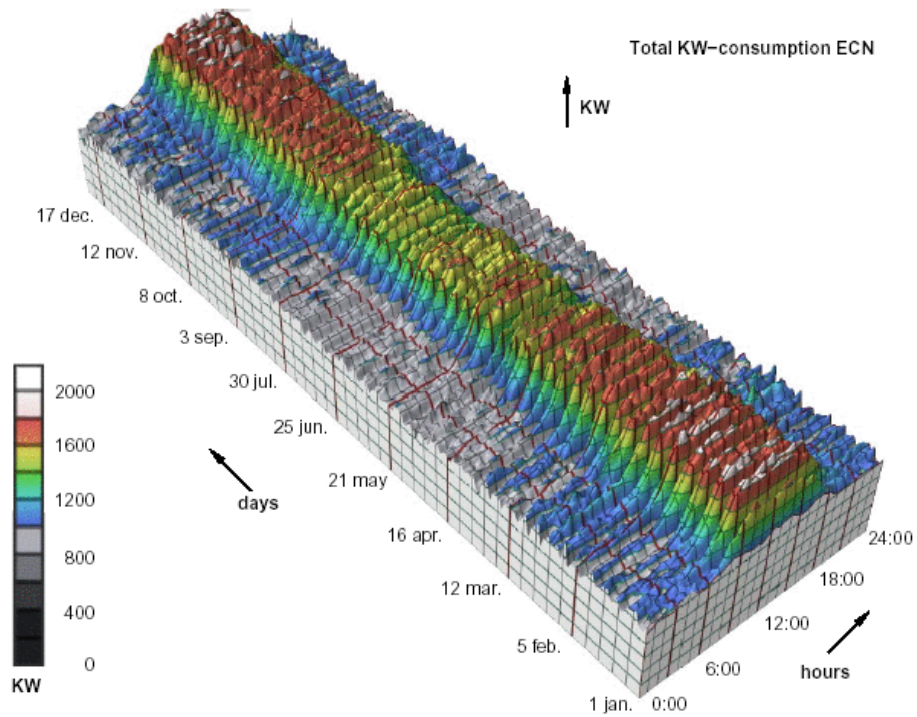
Design study methodology



van Wijk:1999

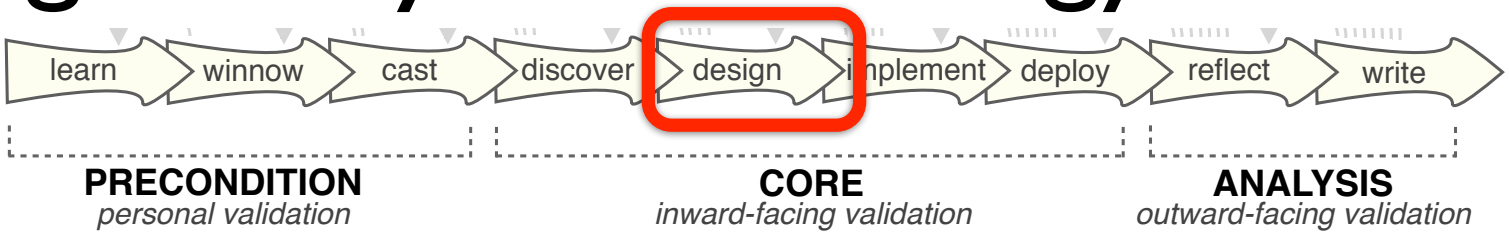
Design study methodology

- Pretty, not so useful
- Daily, weekly patterns are hard to see



van Wijk:1999

Design study methodology



Data abstraction, visual encoding, interaction

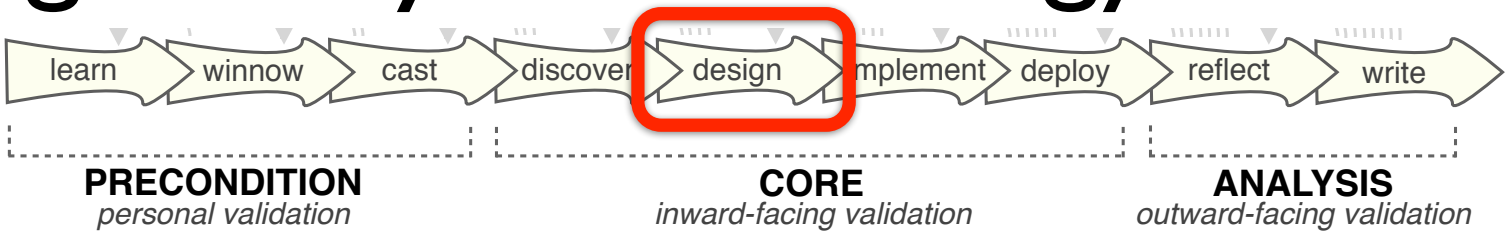
What data transformations are needed?

What visual designs to use?

How to tie this together with interaction?

Don't code!

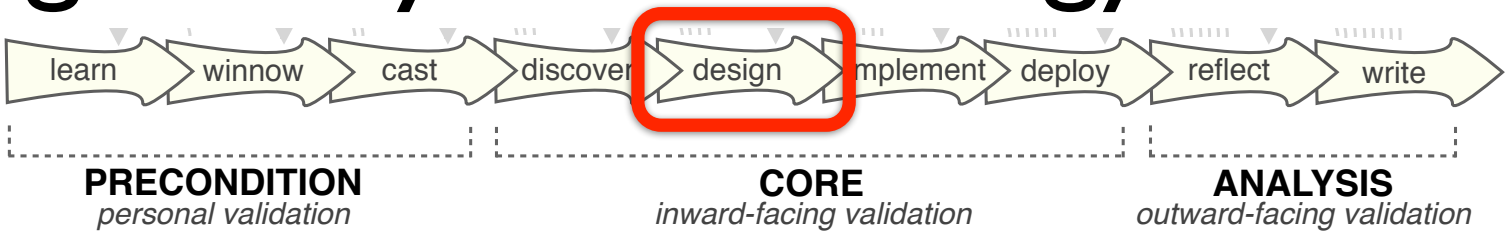
Design study methodology



example: Cluster-Calendar, van Wijk and van Selow

Data transform: hierarchical clustering

Design study methodology



example: Cluster-Calendar, van Wijk and van Selow

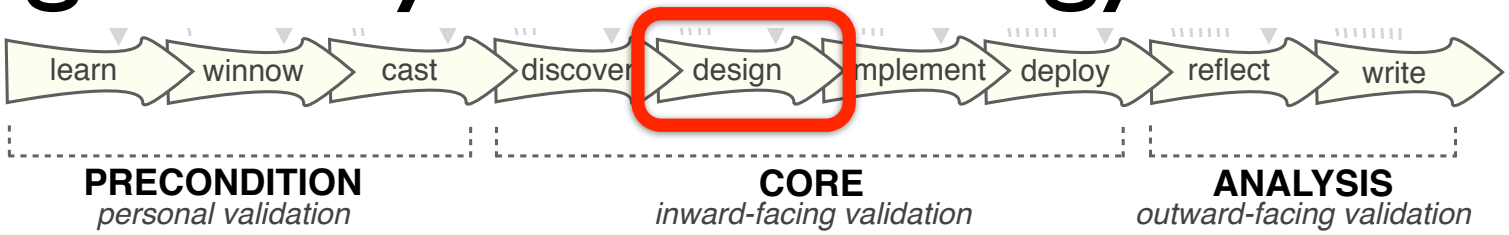
Data transform: hierarchical clustering

start with M day patterns

- compute pair-wise differences, merge most similar
- now we have M-1 patterns
- repeat until we have 1 root cluster

result: binary hierarchy of clusters

Design study methodology



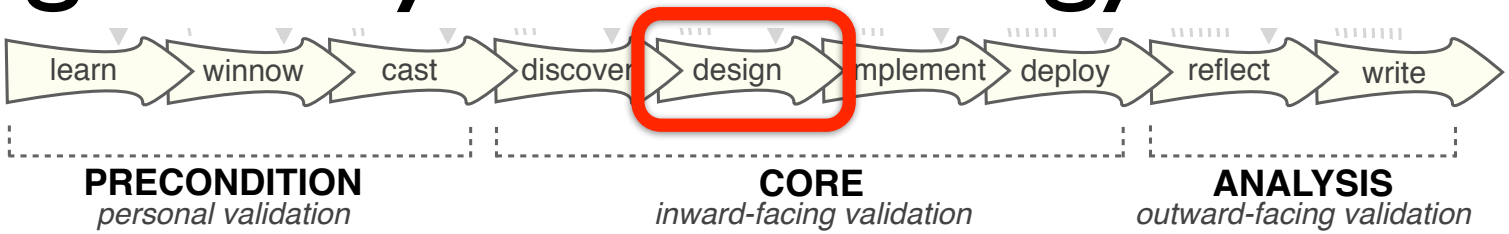
example: Cluster-Calendar, van Wijk and van Selow

Data transform: hierarchical clustering

issues:

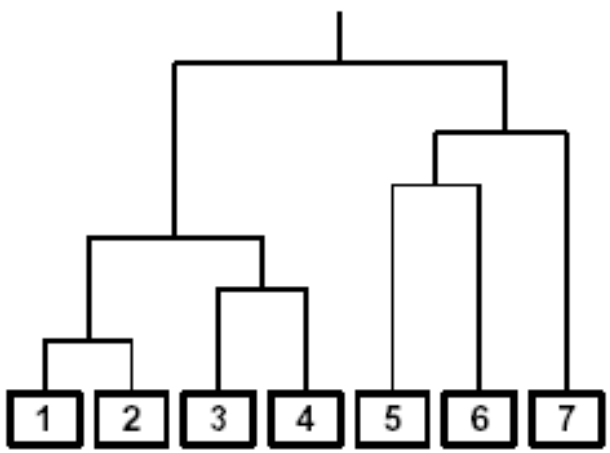
- distance metric to use?
- how to display the cluster?

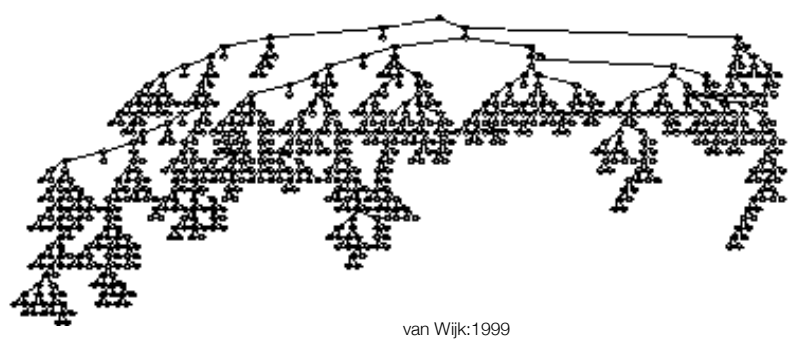
Design study methodology



example: Cluster-Calendar, van Wijk and van Selow

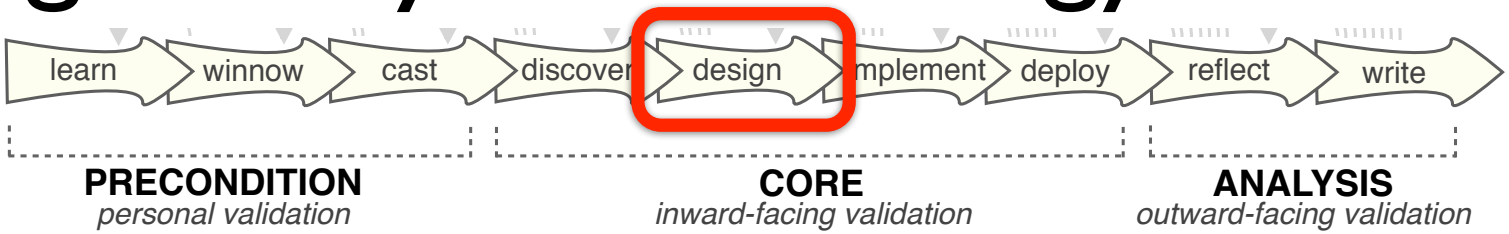
dendrogram



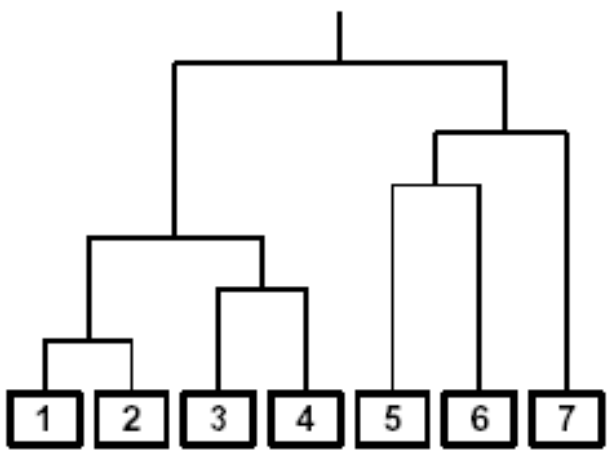


van Wijk:1999

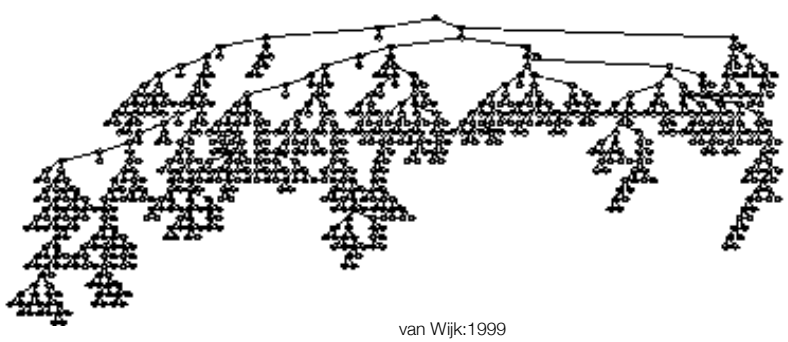
Design study methodology



example: Cluster-Calendar, van Wijk and van Selow

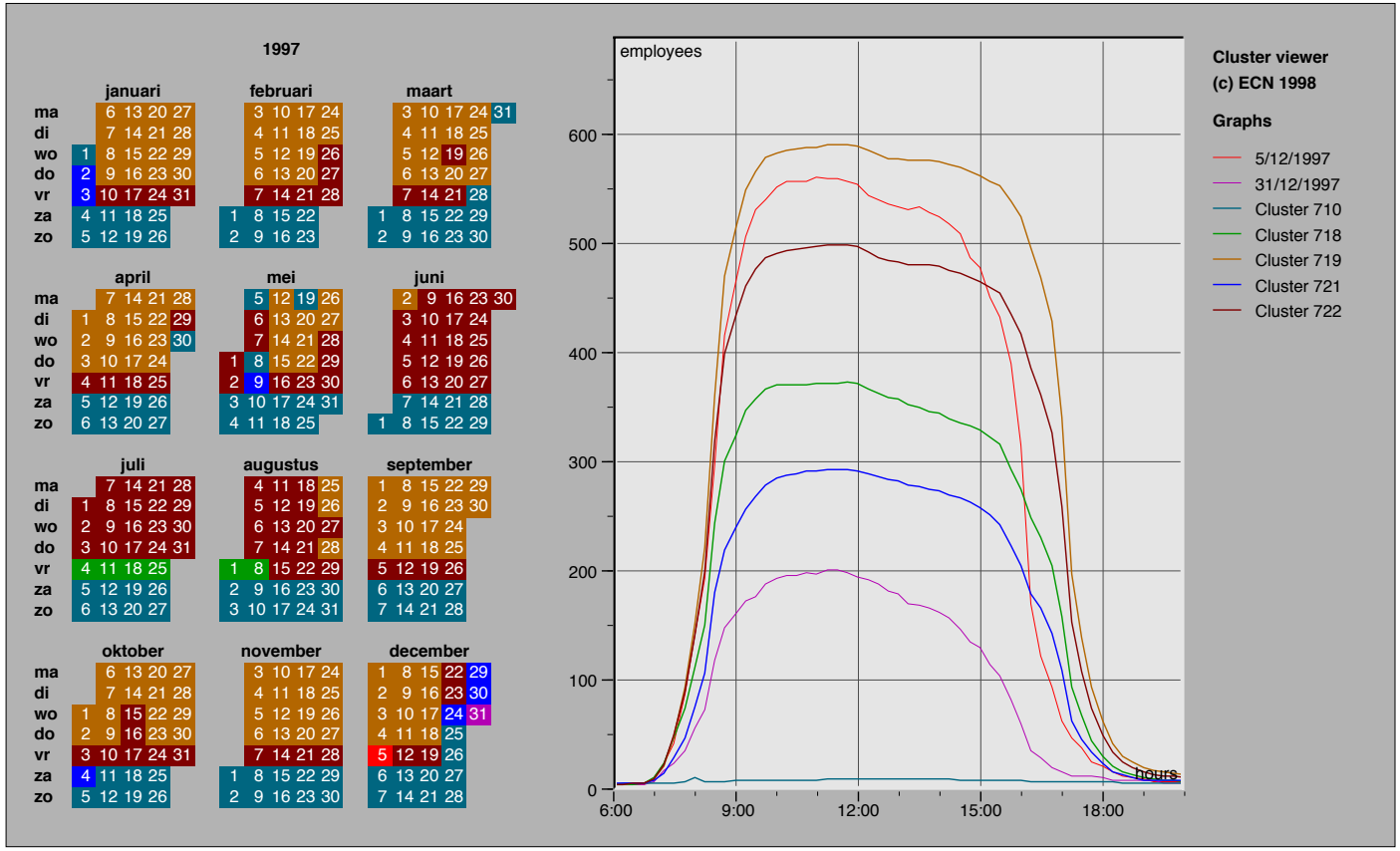


Shows hierarchical structure
but not time distribution!



van Wijk:1999

Design study methodology

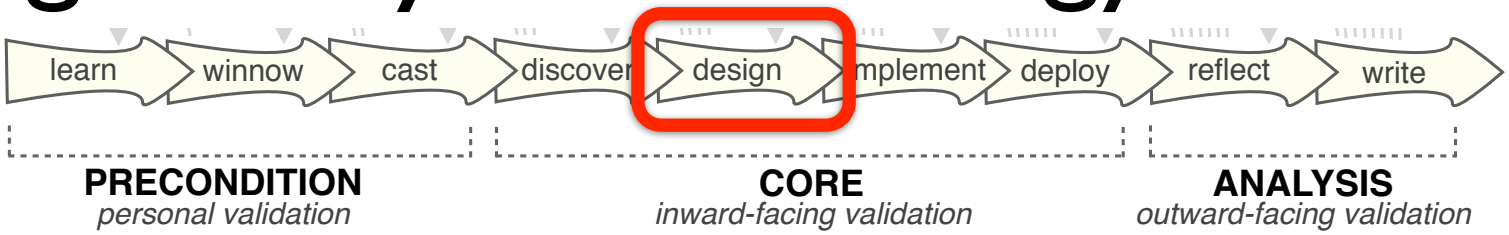


Overview

Detail

van Wijk:1999

Design study methodology



example: Cluster-Calendar, van Wijk and van Selow

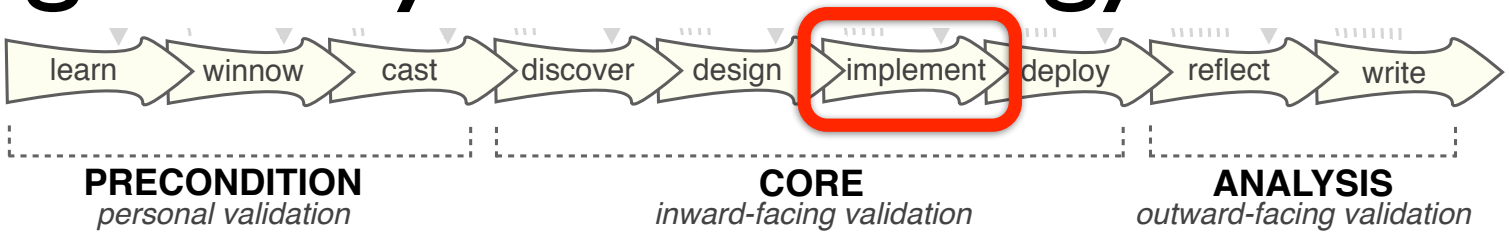
clusters: data transformation to aggregate data

calendar: familiar visual representation for time

linking: interactive exploration of the data

task analysis guided choices: 3D extrusion and dendrogram don't work

Design study methodology



Yay coding!

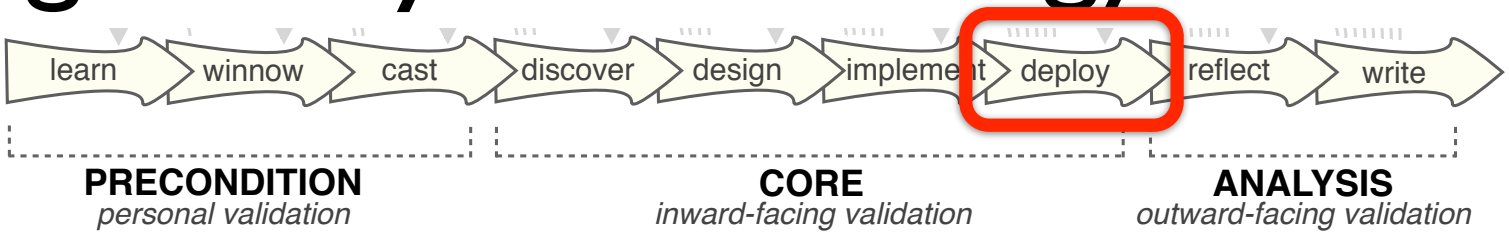
Need to test design hypotheses

Rapid prototyping (will probably throw away alot of code)

Breaking bugs vs annoying bugs

Fast usability testing

Design study methodology



Hand-off to the users

Domain experts need to play with software

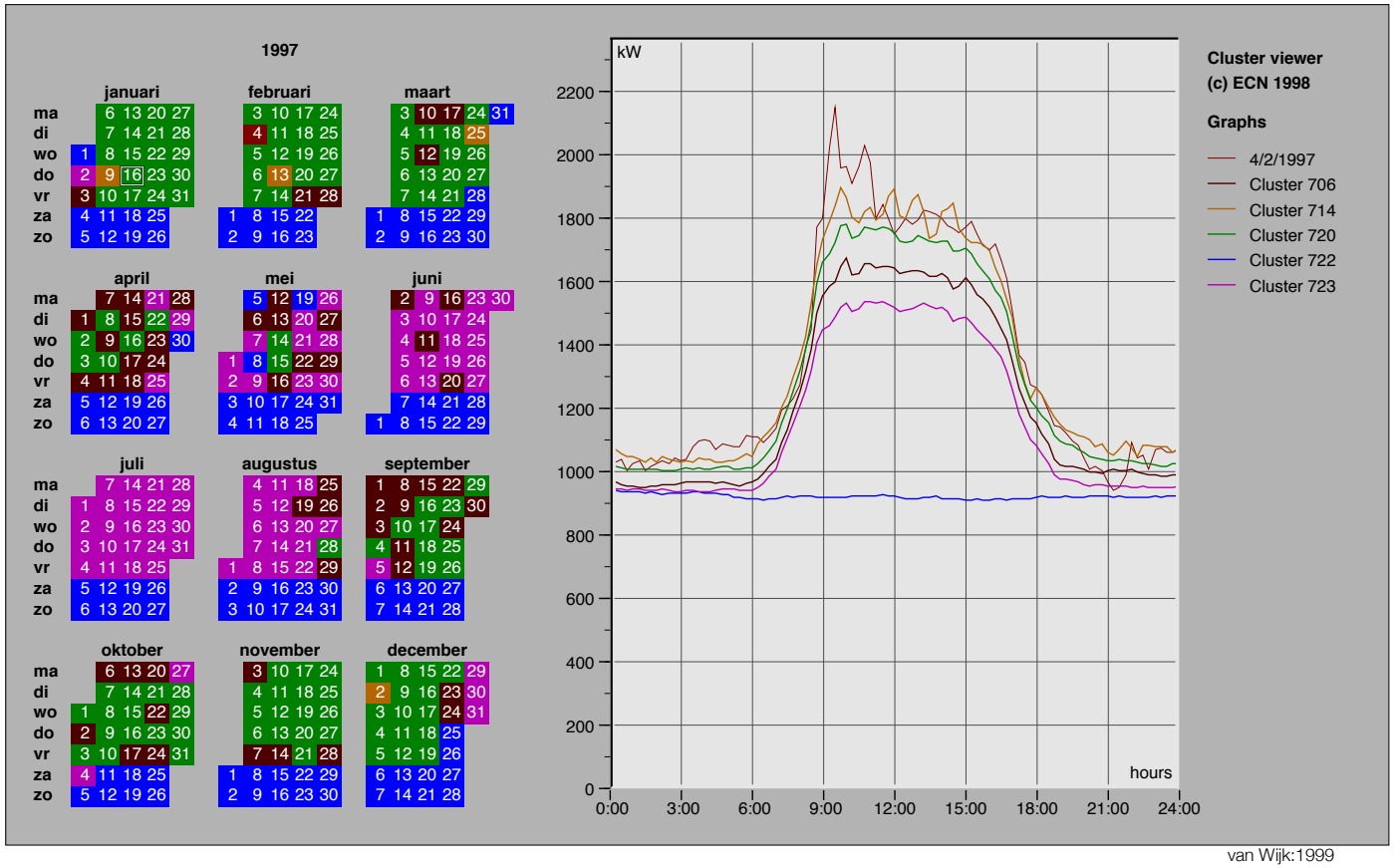
What works, what doesn't?

How to evaluate?

May need to redesign/reimplement a lot

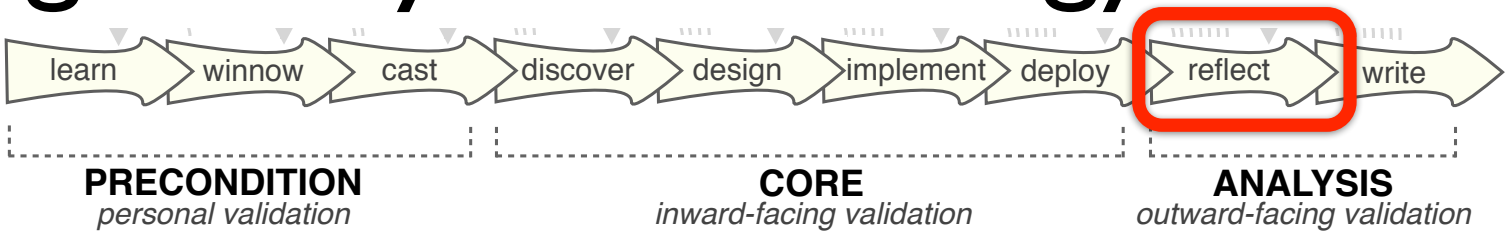
Design study methodology

Critique?



van Wijk:1999

Design study methodology



Refine, reject, propose guidelines

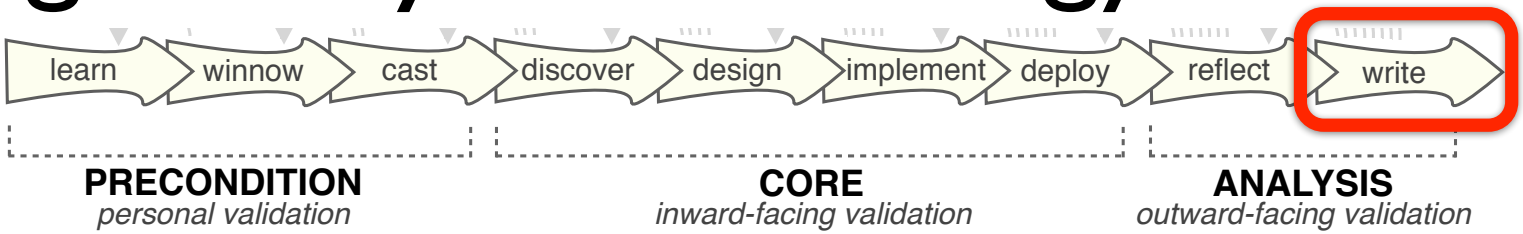
Compare to existing design guidelines

Confirm which ones worked

Reject which ones didn't work

Come up with new guidelines

Design study methodology



Yay words!

Forces clear articulation of problem, tasks, solution

Who else does my study help? - transferability!

Think carefully about what readers will care about

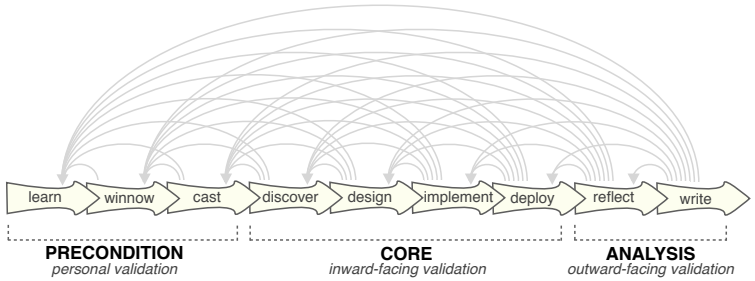
This takes time to do well!

Making the right tool

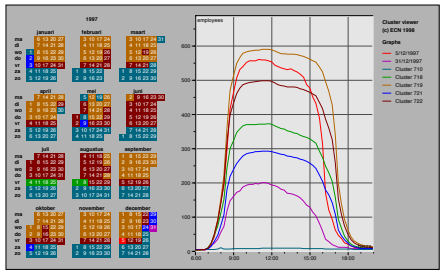
Questions

Data

Tasks



Design study methodology



van Wijk:1999

Where are design studies?

Domain situation
Observe target users using existing tools

Data/task abstraction

Visual encoding/interaction idiom
Justify design with respect to alternatives

Algorithm
Measure system time/memory
Analyze computational complexity

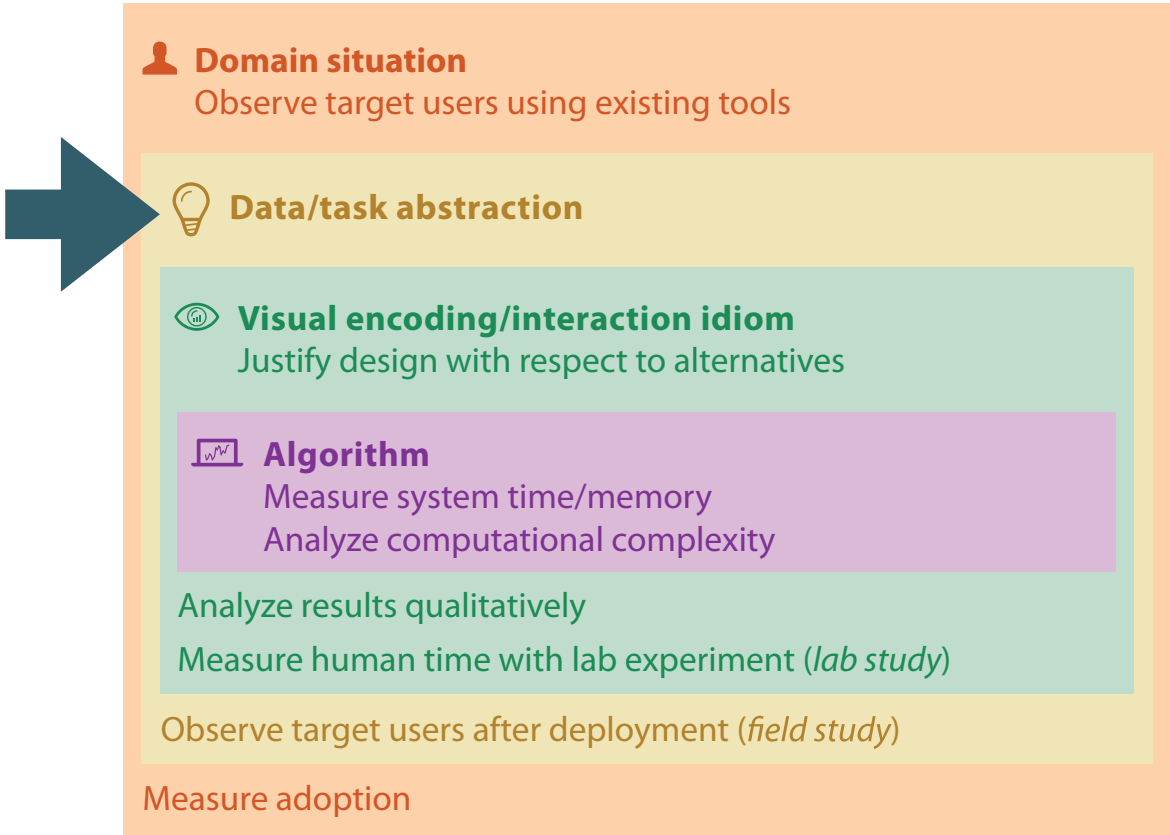
Analyze results qualitatively

Measure human time with lab experiment (*lab study*)

Observe target users after deployment (*field study*)

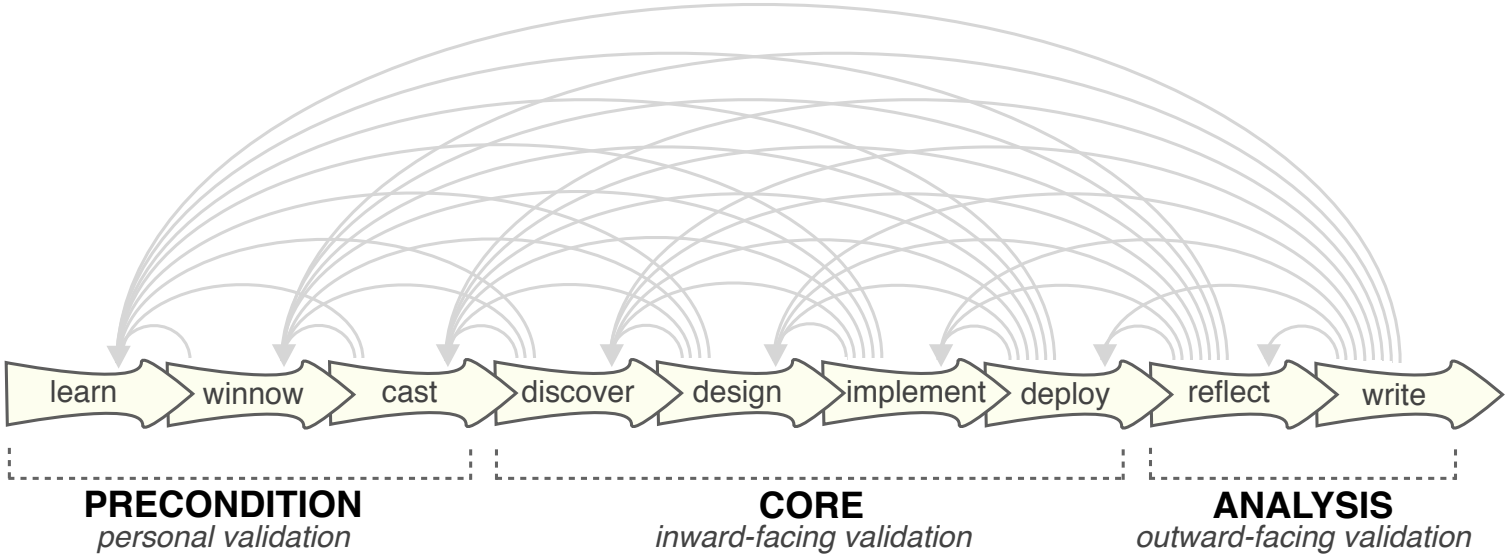
Measure adoption

Where are design studies?



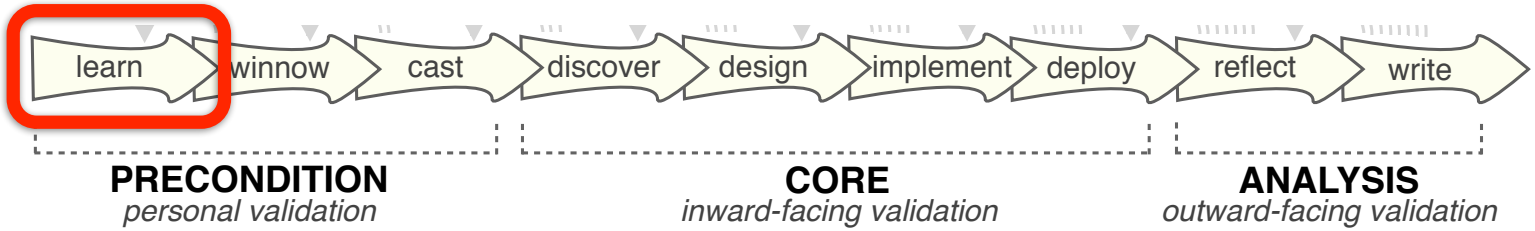
Pitfalls

Pitfalls



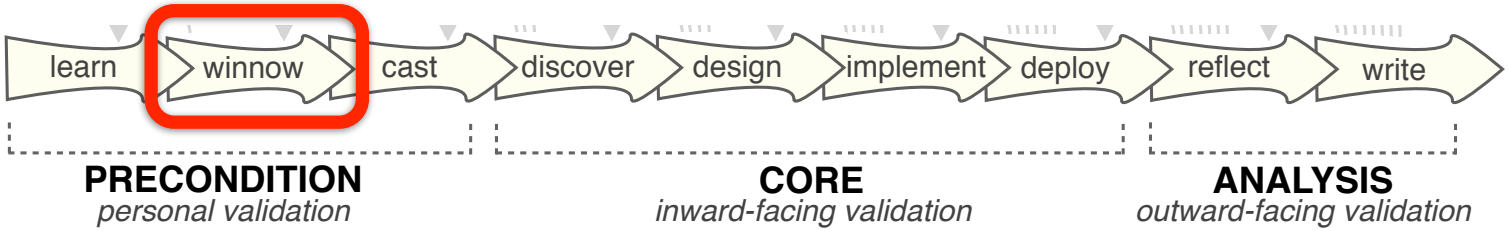
#1: Don’t skip steps!

Pitfalls



insufficient knowledge of literature

Pitfalls



collaboration with the wrong people

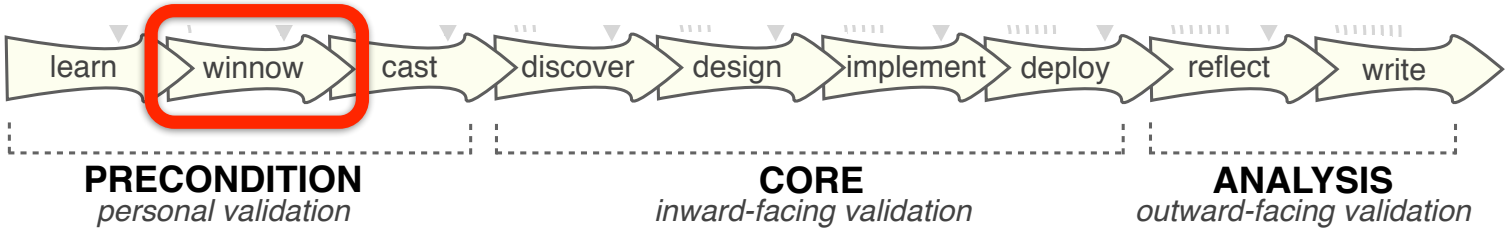
no real data available

insufficient time available from collaborators

no need for visualization: automate

no need for research: engineering project

Pitfalls



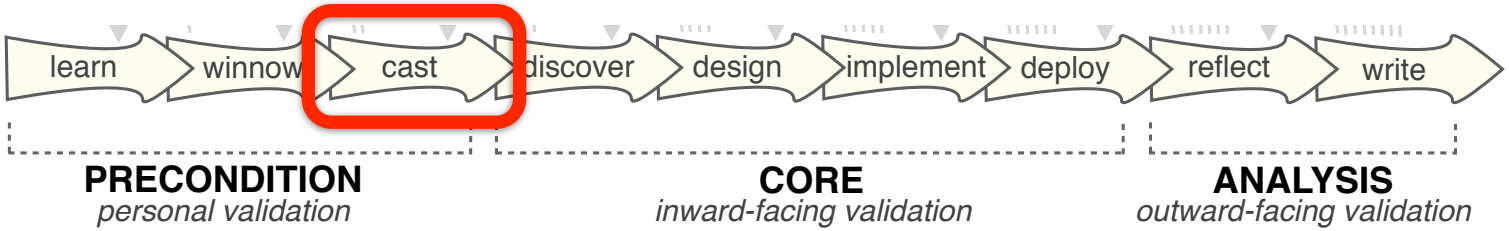
is this interesting to me?

existing tools are good enough

not an important/recurring task

no rapport with collaborators

Pitfalls

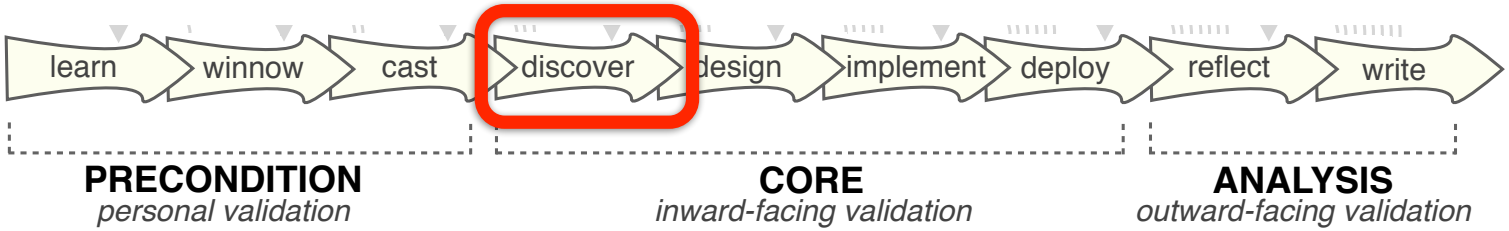


not identifying front-line analyst and gatekeeper

assuming same role distribution across projects

mistaking tool-builders for real end users

Pitfalls



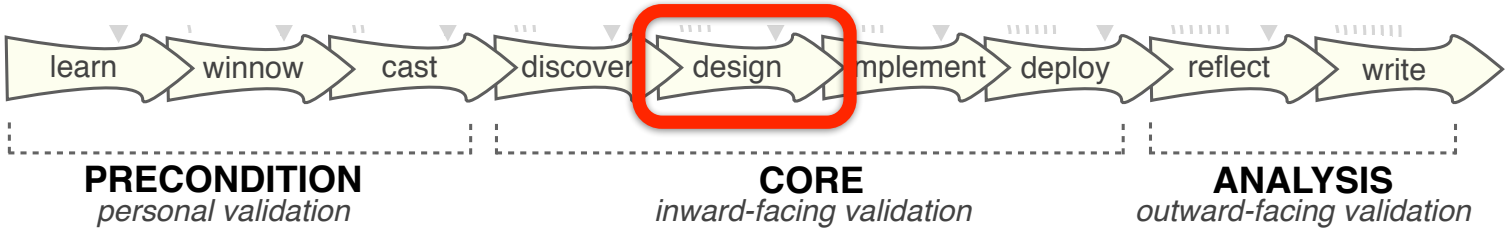
ignoring practices that currently work well

expecting *just talking* or *fly on the wall* to work

domain experts design the visualizations

too much/too little domain knowledge

Pitfalls

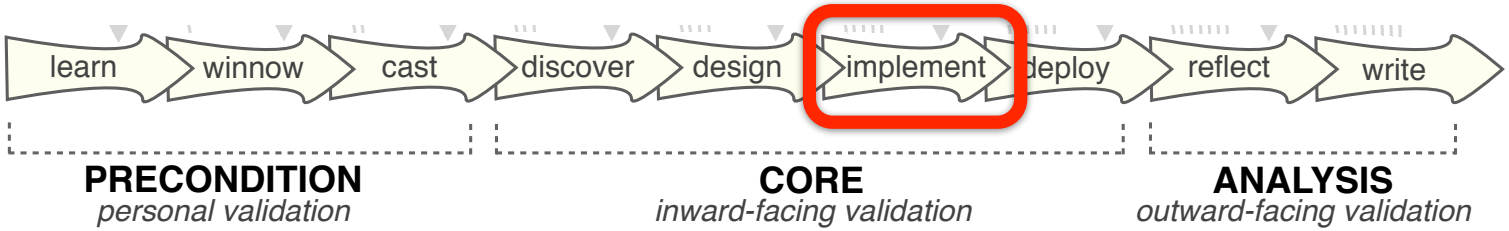


too little abstraction

design consideration space too small

mistaking technique-driven and problem-driven work

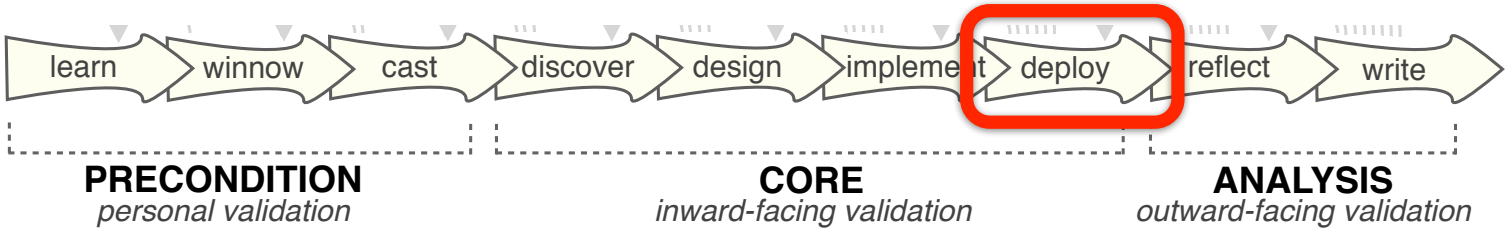
Pitfalls



non-rapid prototyping

usability: too little/too much

Pitfalls



insufficient deploy time

non-real task/data/user

liking a tool is not validation!