

Another Challenge: Staffing Turmoil

- Your project needs to **introduce more competitive features** and you're hiring additional developers
- Your **competitors' products are more reliable**
- You're hiring **additional developers** to find/fix defects
- Your **company is growing** and your developers have accepted promotions to new jobs, and you're hiring new developers to replace them
- **How will you introduce your new developers to your project?**

Another Challenge: Dispersed Teams

- What if management staffed the *War and Peace* project at **multiple locations**:
 - Different floors of the same building
 - Different building of the same campus
 - Different countries/continents
- How will that complicate development?

Agility: An Emerging Motivation

- Agility refers to our response to changing project conditions
 - Customers don't know **what they need**
 - The customers' **needs change**
 - Competitive pressure forces unexpected **change**
 - Development is more **difficult** than expected
 - Technologies **change**
 - Personnel **changes**
 - Disaster: Equipment failure, weather, fire...
- Business managers can leverage agility to create value (\$\$\$)

Teamwork Arises from Competitive Pressure

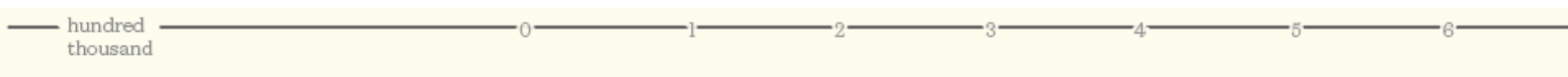
- Most software is large: tens, hundreds, even millions of lines of code (LOC)
- 1 KLOC = ... LOC
- 1 MLOC = ... KLOC = ... LOC

Teamwork Arises from Competitive Pressure

- Most software is large: tens, hundreds, even millions of lines of code (LOC)
- 1 KLOC = 1,000 LOC
- 1 MLOC = 1,000 KLOC = 1,000,000 LOC

Codebases

Hundred thousand
= 100 KLOC
= 100,000 LOC

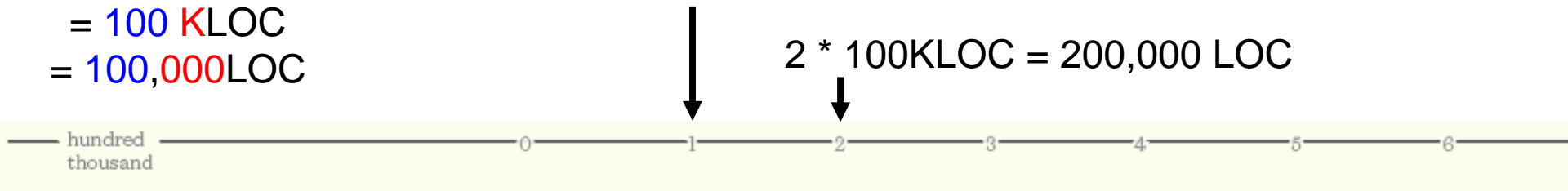


Codebases

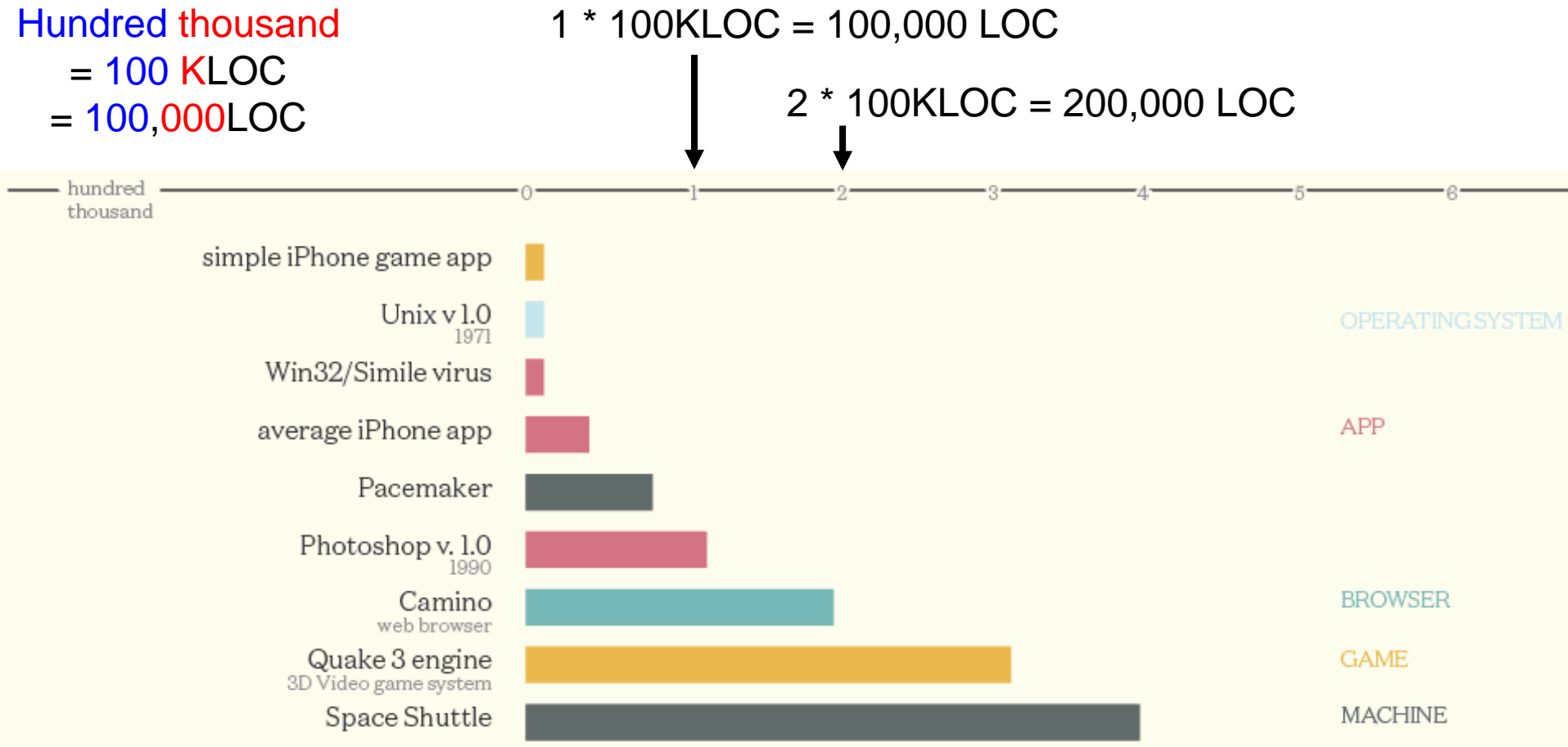
Hundred thousand
= 100 KLOC
= 100,000 LOC

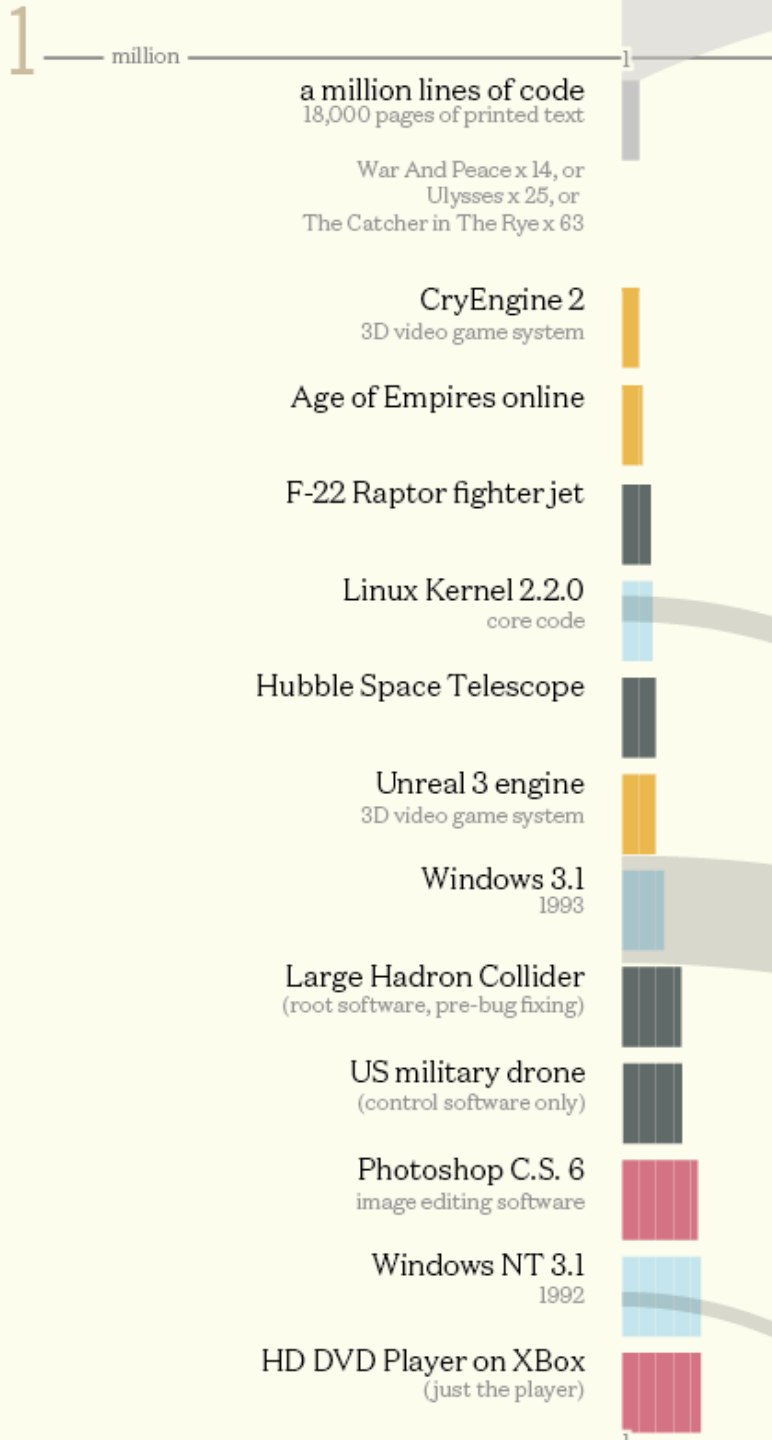
$$1 * 100\text{KLOC} = 100,000 \text{ LOC}$$

$$2 * 100\text{KLOC} = 200,000 \text{ LOC}$$



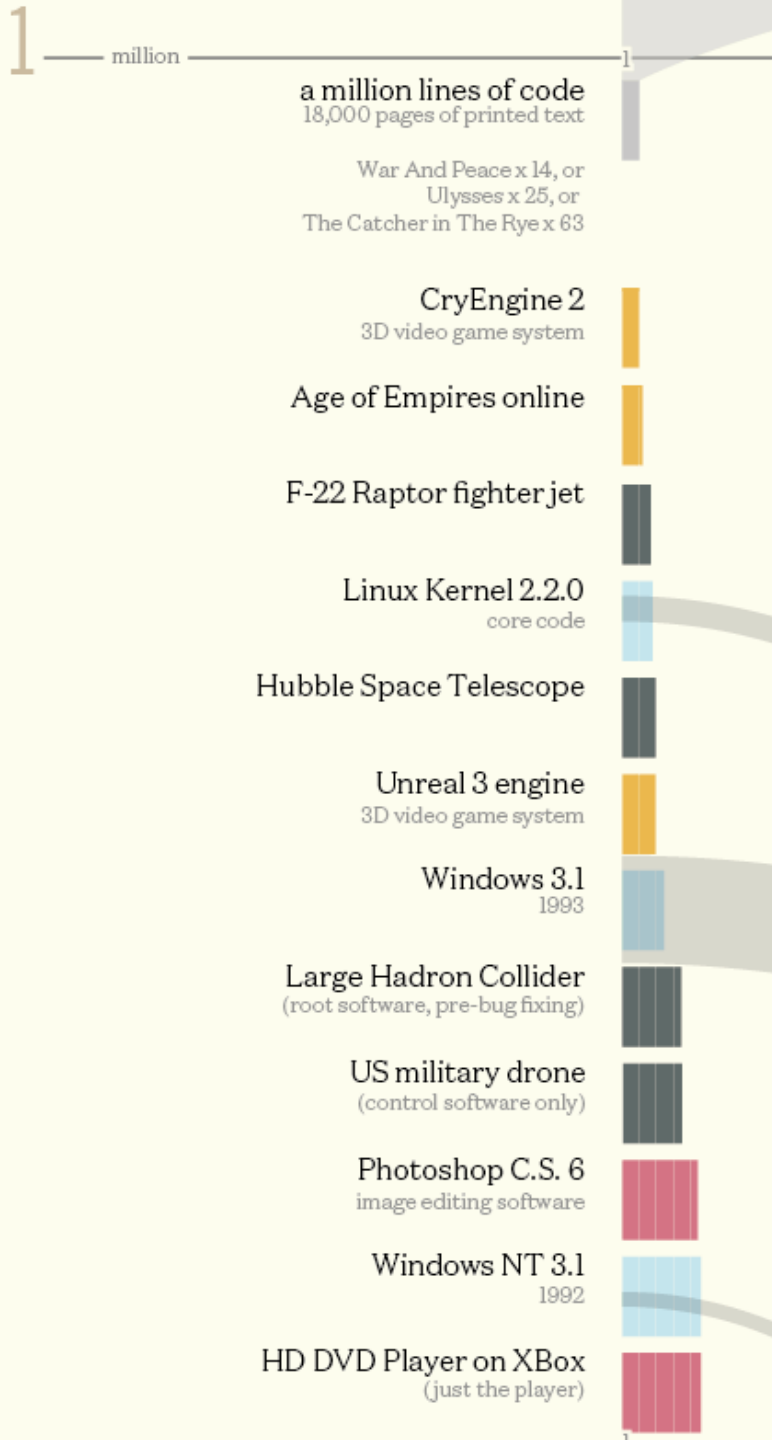
Codebases





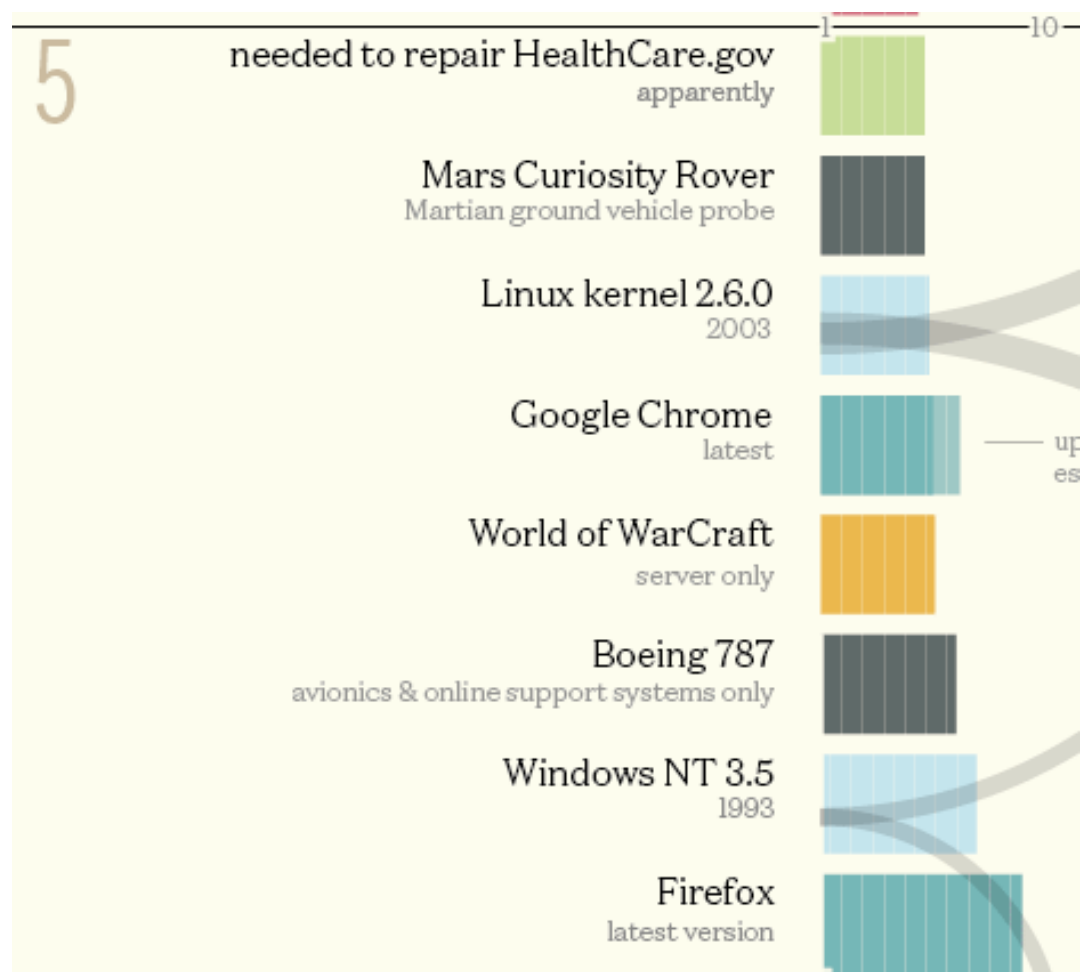
Codebases (MLOC)

<http://www.informationisbeautiful.net/visualizations/million-lines-of-code/>

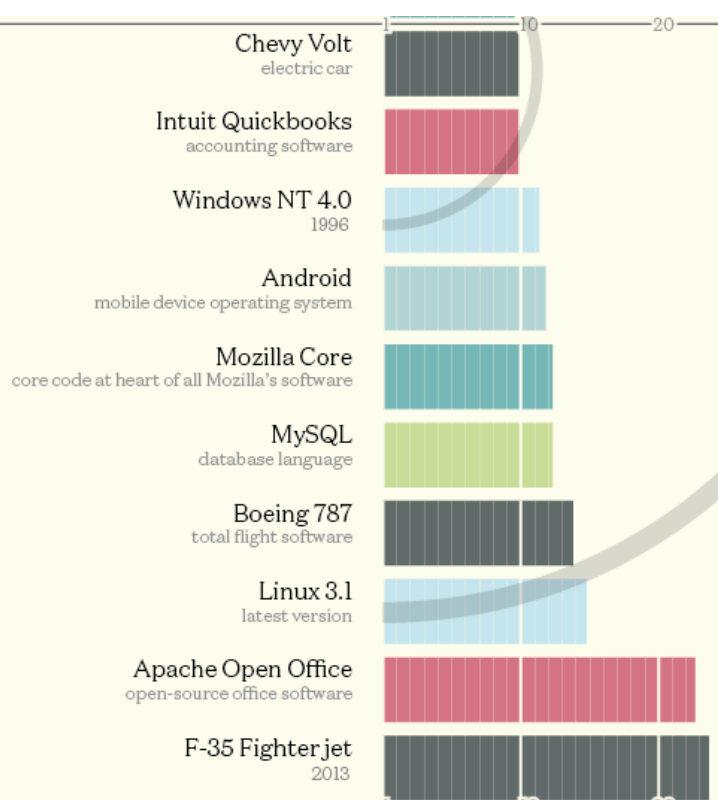


Codebases (MLOC)

<http://www.informationisbeautiful.net/visualizations/million-lines-of-code/>



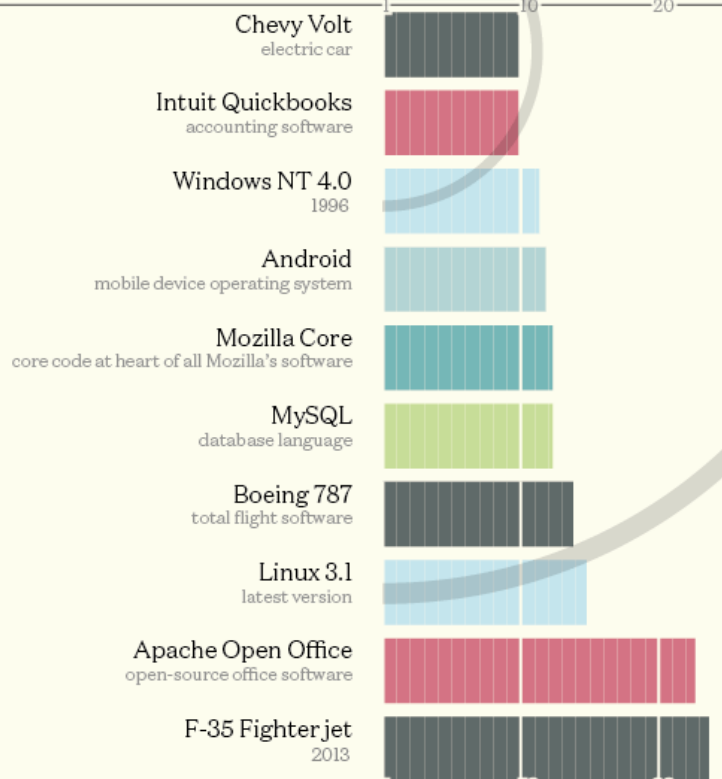
10



Codebases (MLOC)

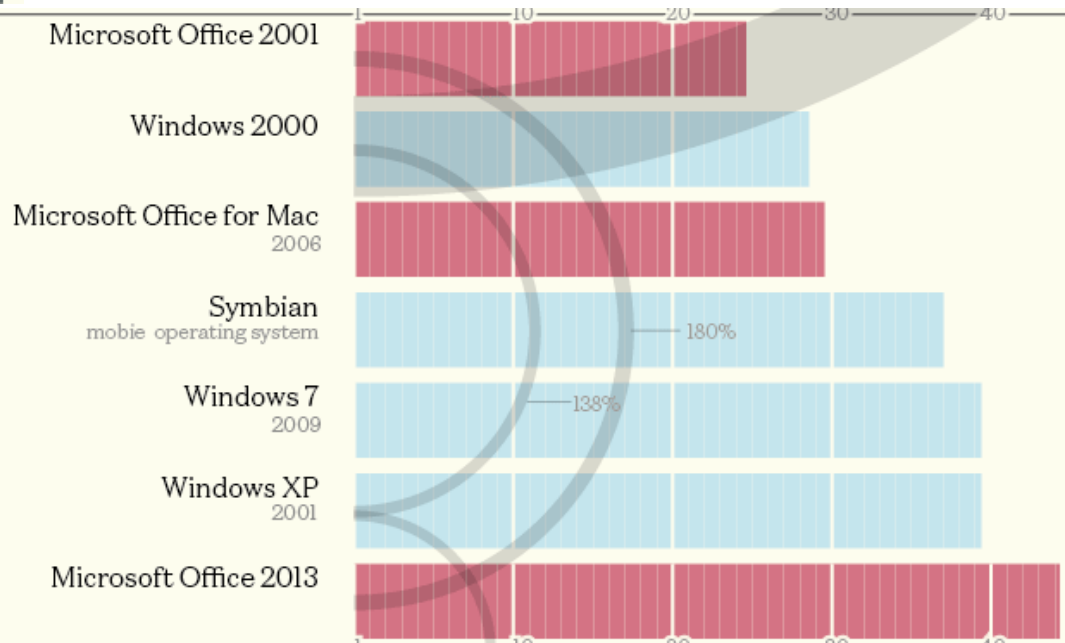
<http://www.informationisbeautiful.net/visualizations/million-lines-of-code/>

10



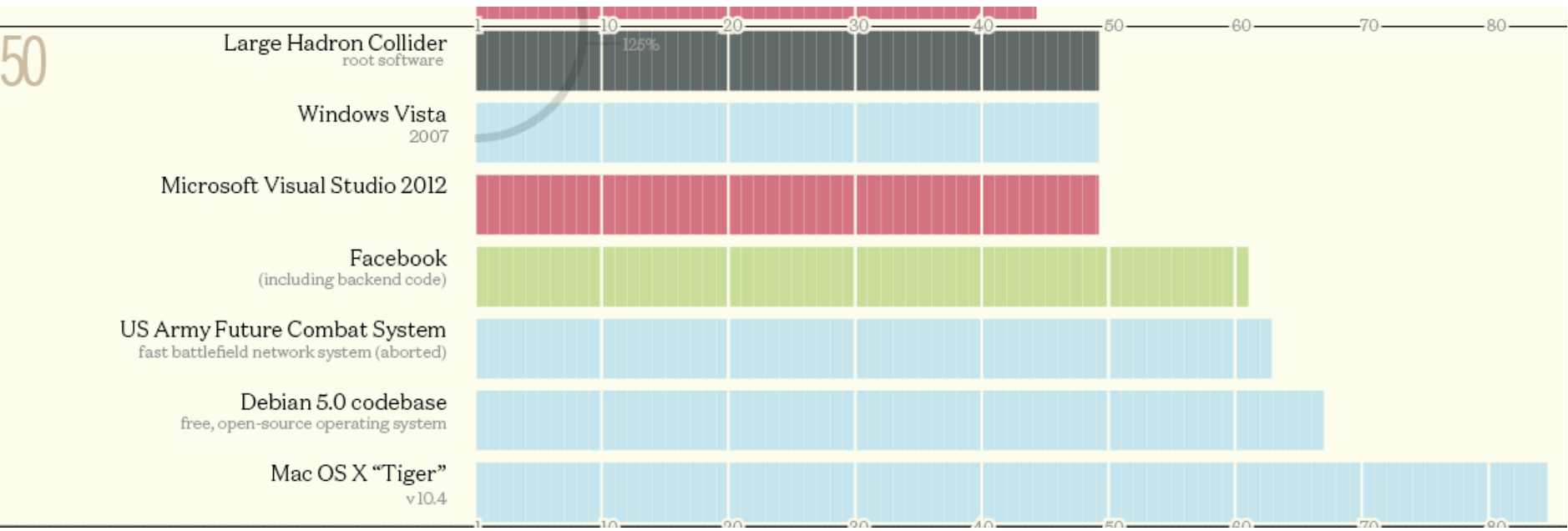
Codebases (MLOC)

25

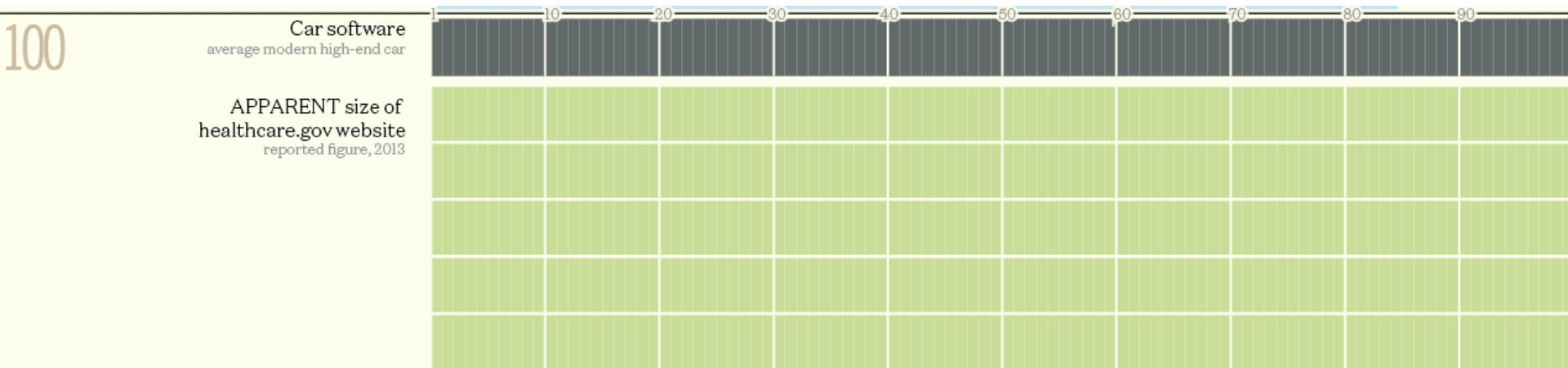


<http://www.informationisbeautiful.net/visualizations/million-lines-of-code/>

Codebases (MLOC)



Codebases (MLOC)



Google Codebase Size?

Google Codebase Size?

- 2 billion LOC

Teamwork Arises from Competitive Pressure

- Most software is large: tens, hundreds, even millions of lines of code (LOC)

Teamwork Arises from Competitive Pressure

- Most software is large: tens, hundreds, even millions of lines of code (LOC)
- Business can't wait for you to develop all that code alone
- You will work in a team
- Some of you will work in a network of teams, some who you will never meet in person (think... Open Source development)

Incremental Development

Incremental Development

- Product is constructed in incremental releases rather than a single release
 - Releases, but not activities, are usually gated
 - what do we mean by gated?

Incremental Development

- Product is constructed in incremental releases rather than a single release
 - Releases, but not activities, are usually gated (i.e., new changes are not allowed for the current release)
- Each release incorporates a set of activities
 - Requirements
 - Design
 - Implementation
 - Testing, etc.

Incremental Development

- Initial release often delivers a very minimal experience (e.g., User Interface with nothing behind it)
- Subsequent releases deliver increasing functionality

Incremental Development Benefits

- (from the client/customer point of view...)

Incremental Development Benefits

- Agility reduces the cost of responding to change
 - The amount of analysis and documentation that has to be redone is much less than is required with the waterfall model
 - Change is “expected”
- Customers validate incremental releases
 - Customers comment on use or demonstrations of the software and see how much has been implemented
 - Project often identifies and recovers from problems early

Incremental Development Benefits (contd)

- More rapid delivery and deployment of useful software to the customer is possible
 - Customers use and gain value from early releases
 - “First to market” advantage
 - Features can be implemented in prioritized order

Incremental Development Limitations

Incremental Development Limitations

- The software design **rots without regular refactoring**, making **future changes more expensive** over what becomes a legacy design
 - **Solution:** Routinely include “code health” activities and engineering practices

Incremental Development Practice

- **Scrum** is the most widely used incremental development process
 - We'll spend several weeks on it

Waterfall vs. Incremental/Agile (CHAOS report from Standish Group)

CHAOS report from Standish Group

- between 1994-2004: 66% - 84% of software projects were either:

CHAOS report from Standish Group

- between 1994-2004: 66% - 84% of software projects were either:
 - canceled before completion or completed late,
 - over budget, or
 - with features missing
- Large percentage of failures

2015 CHAOS report from Standish Group

- 50K projects studied

2015 CHAOS report from Standish Group

- 50K projects studied

MODERN RESOLUTION FOR ALL PROJECTS

	2011	2012	2013	2014	2015
SUCCESSFUL	29%	27%	31%	28%	29%
CHALLENGED	49%	56%	50%	55%	52%
FAILED	22%	17%	19%	17%	19%

The Modern Resolution (OnTime, OnBudget, with a satisfactory result) of all software projects from FY2011 - 2015 within the new CHAOS database. Please note that for the rest of this report CHAOS Resolution will refer to the Modern Resolution definition not the Traditional Resolution definition.

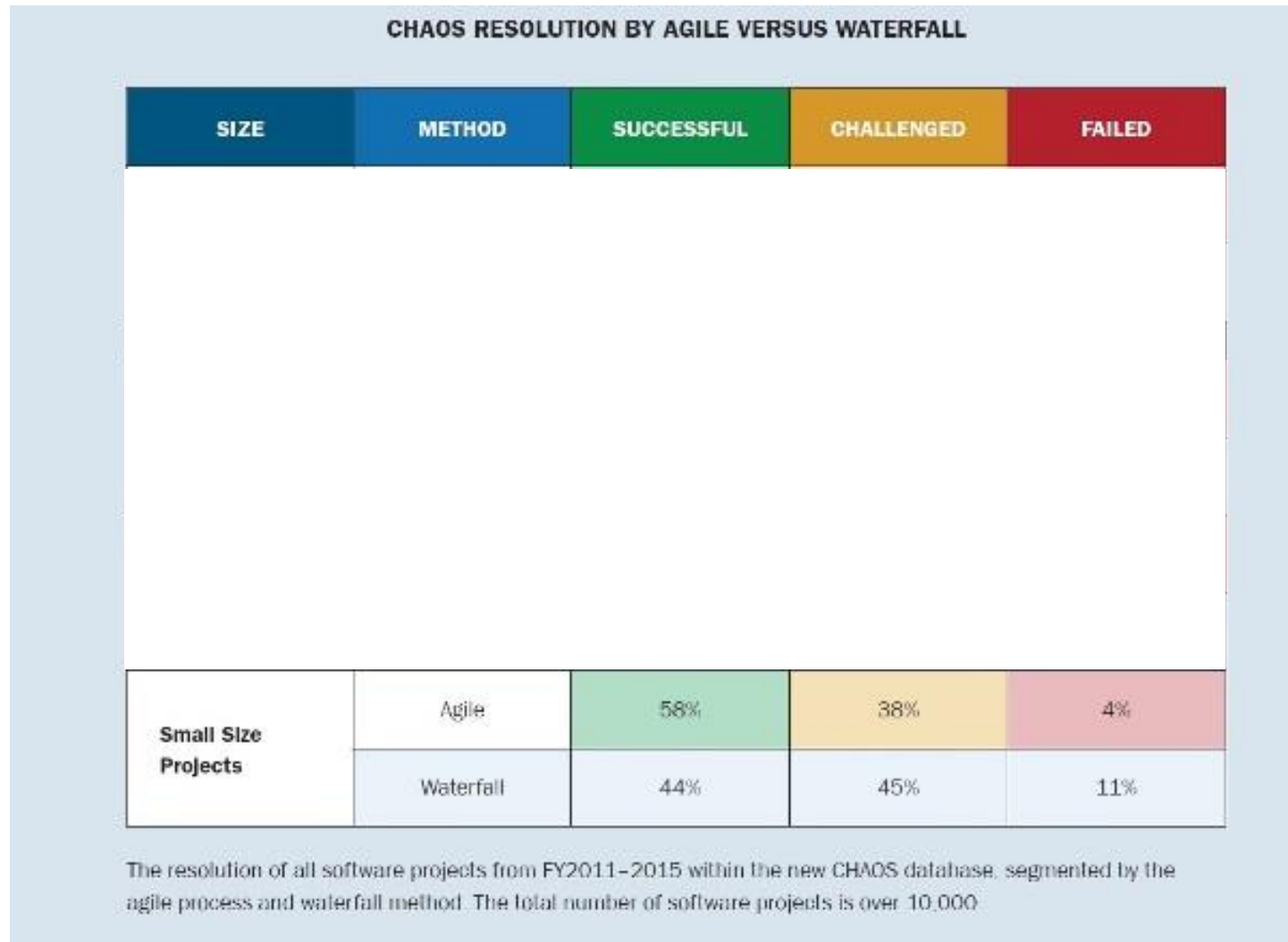
2015 CHAOS report from Standish Group

CHAOS RESOLUTION BY PROJECT SIZE

	SUCCESSFUL	CHALLENGED	FAILED
Grand	2%	7%	17%
Large	6%	17%	24%
Medium	9%	26%	31%
Moderate	21%	32%	17%
Small	62%	16%	11%
TOTAL	100%	100%	100%

The resolution of all software projects by size from FY2011–2015 within the new CHAOS database.

2015 CHAOS report from Standish Group



2015 CHAOS report from Standish Group

CHAOS RESOLUTION BY AGILE VERSUS WATERFALL

SIZE	METHOD	SUCCESSFUL	CHALLENGED	FAILED
Medium Size Projects	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Small Size Projects	Agile	58%	38%	4%
	Waterfall	44%	45%	11%

The resolution of all software projects from FY2011-2015 within the new CHAOS database, segmented by the agile process and waterfall method. The total number of software projects is over 10,000.

2015 CHAOS report from Standish Group

CHAOS RESOLUTION BY AGILE VERSUS WATERFALL

SIZE	METHOD	SUCCESSFUL	CHALLENGED	FAILED
Large Size Projects	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Medium Size Projects	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Small Size Projects	Agile	58%	38%	4%
	Waterfall	44%	45%	11%

The resolution of all software projects from FY2011–2015 within the new CHAOS database, segmented by the agile process and waterfall method. The total number of software projects is over 10,000.

2015 CHAOS report from Standish Group

CHAOS RESOLUTION BY AGILE VERSUS WATERFALL

SIZE	METHOD	SUCCESSFUL	CHALLENGED	FAILED
All Size Projects	Agile	39%	52%	9%
	Waterfall	11%	60%	29%
Large Size Projects	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Medium Size Projects	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Small Size Projects	Agile	58%	38%	4%
	Waterfall	44%	45%	11%

The resolution of all software projects from FY2011–2015 within the new CHAOS database, segmented by the agile process and waterfall method. The total number of software projects is over 10,000.