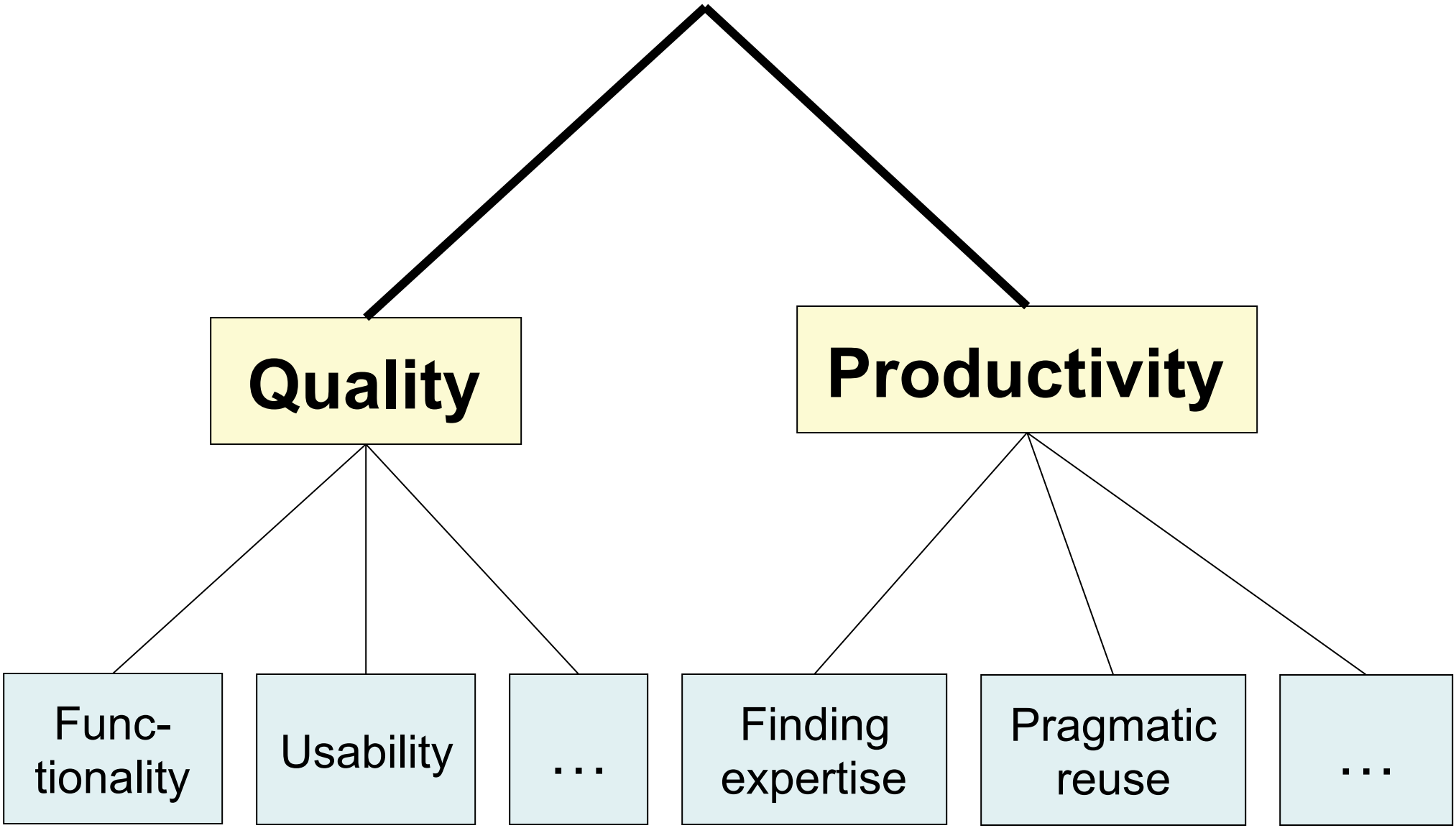
Studying the Social Aspects of Software Development

Nan Niu

Department of Electrical Eng. & Computing Systems
University of Cincinnati
http://homepages.uc.edu/~niunn

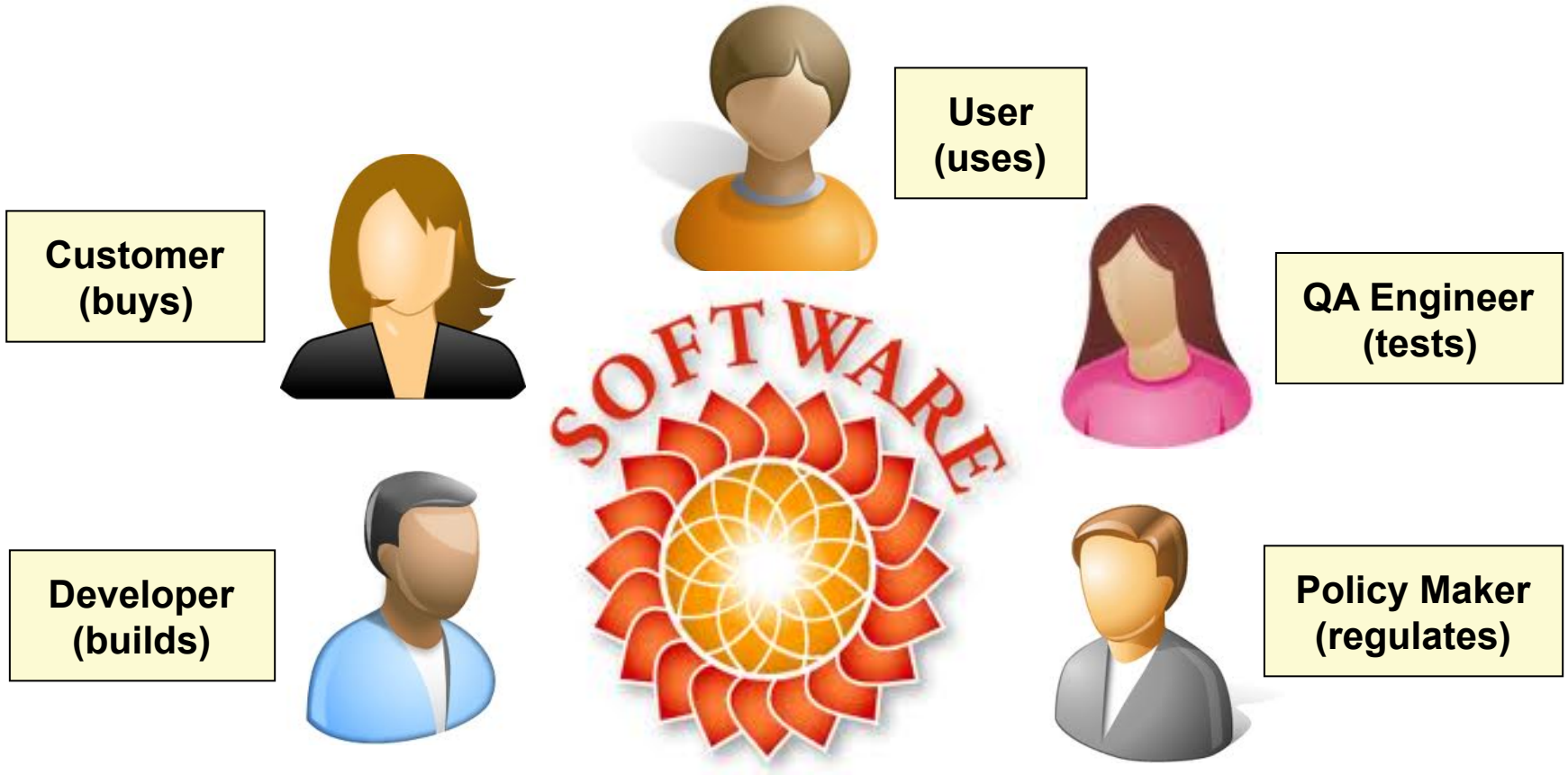
December 5, 2016 @ DIANA/HEP

Software Engineering



Software ≈ Peopleware

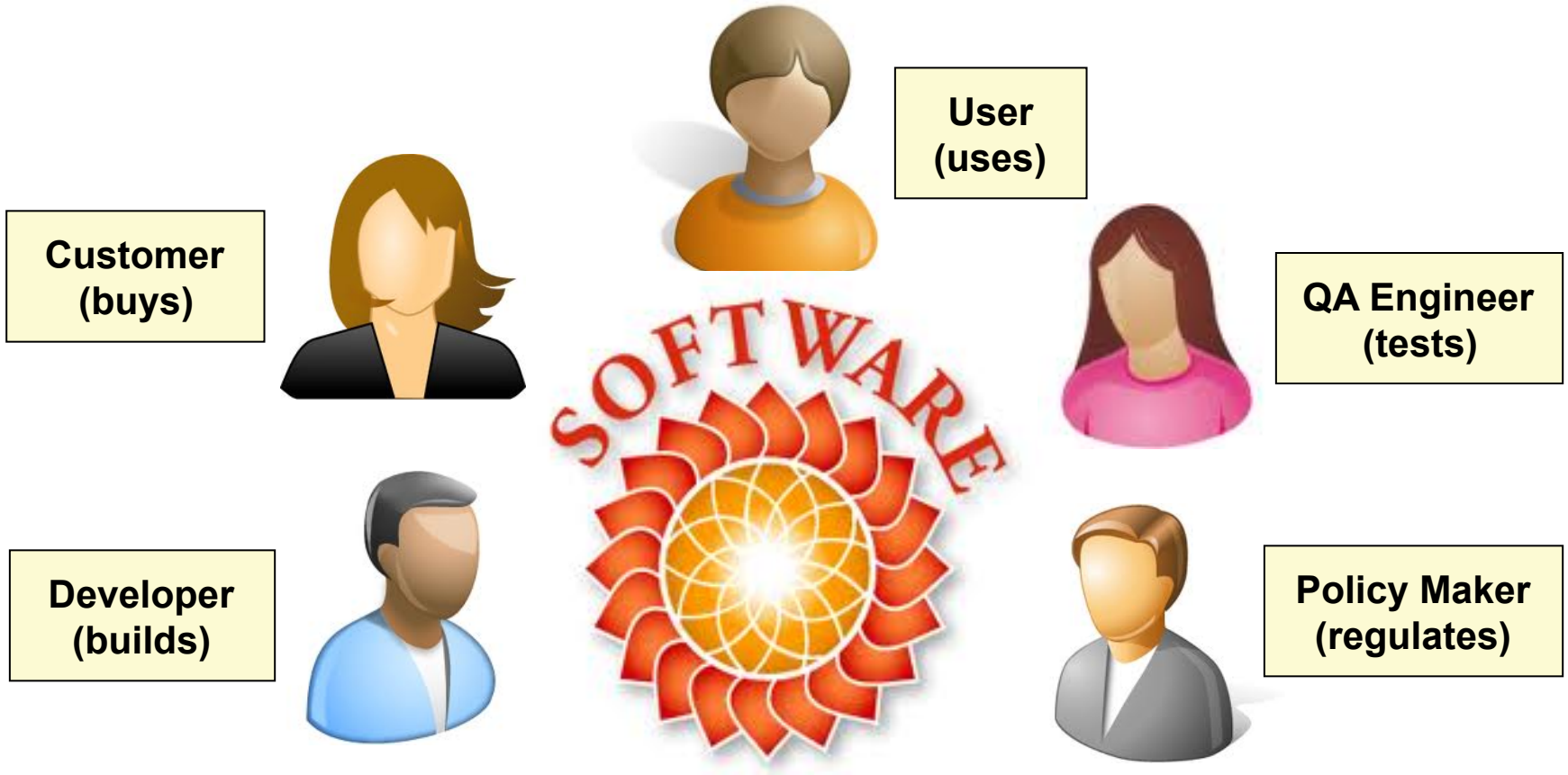
[DeMarco and Lister, 1987]



Source: Google Images (applies to all images in this presentation except otherwise noticed)

Software ≈ Peopleware

[DeMarco and Lister, 1987]

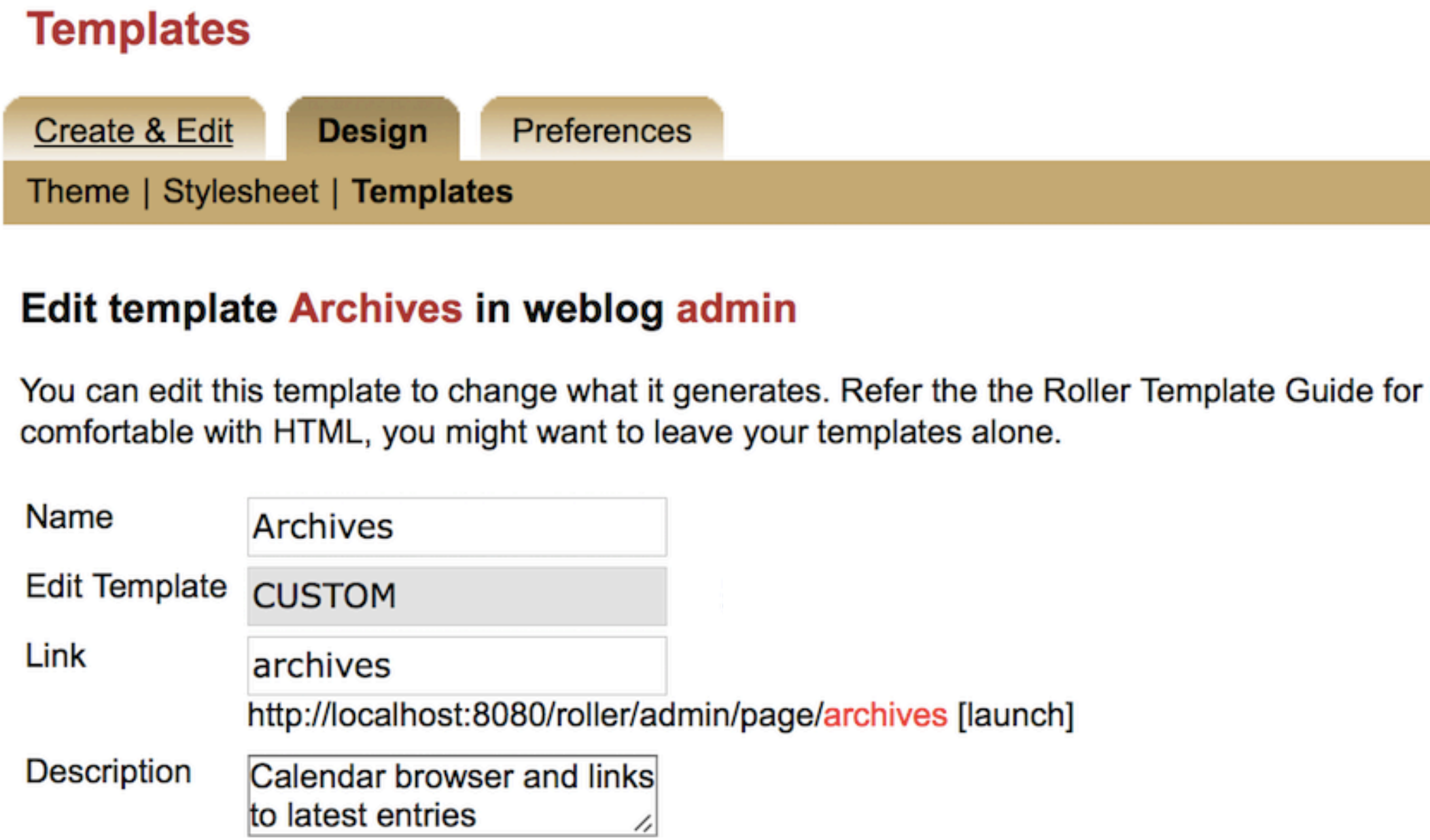


"Software is not limited by physics, but by properties of people."

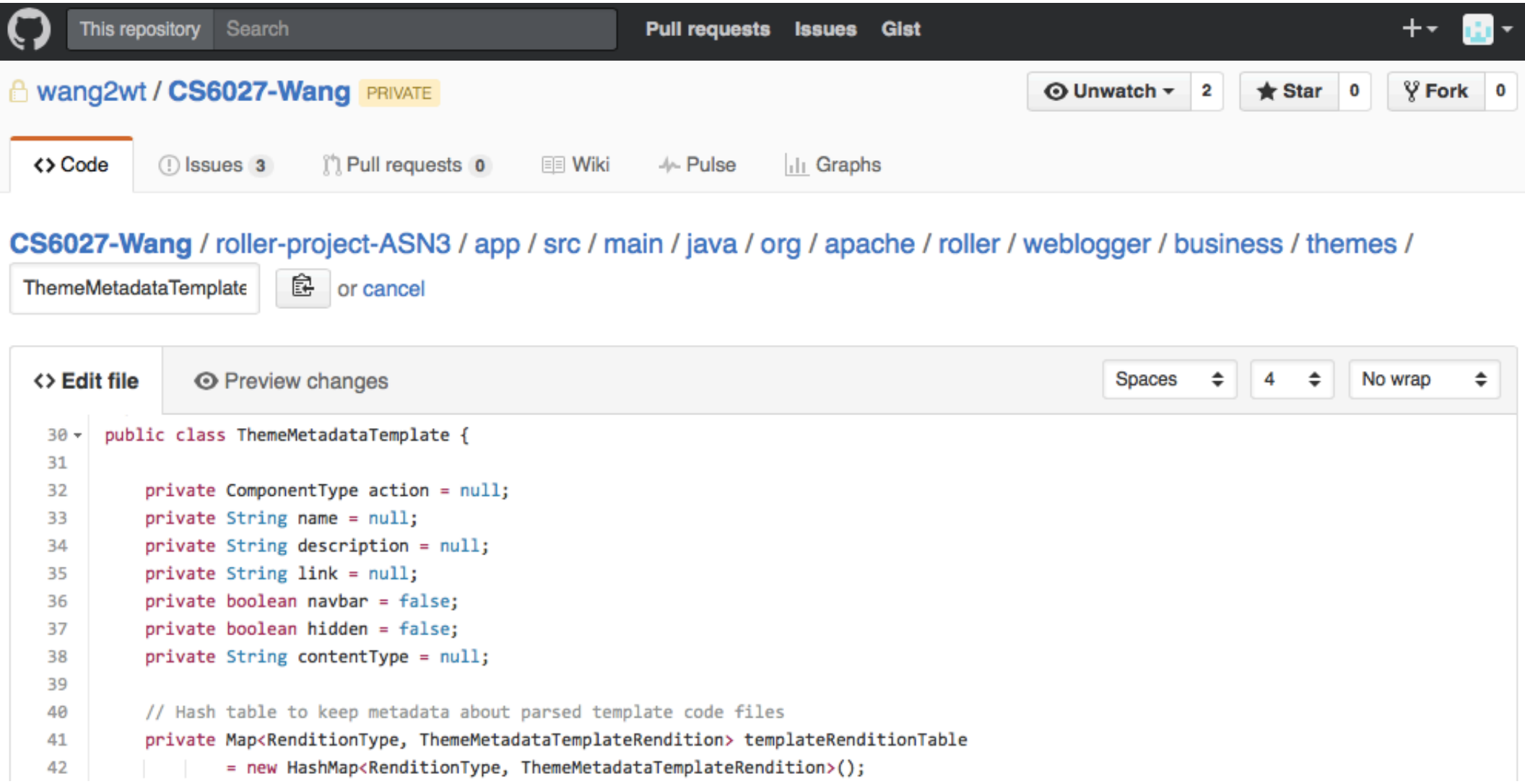
Ralph Johnson

Source: Google Images (applies to all images in this presentation except otherwise noticed)

Human as User



Human as Builder



What is Engineering?

- Creating cost-effective solutions
  - Not just about solving problems, but doing so with economical use of all resources
- by applying scientific knowledge
  - Solving problems in a particular way: by applying science, mathematics, and design analysis
- to build things
  - Emphasizing the solutions, which are usually tangible artifacts
- in the service of mankind
  - Not only serving the immediate customer, but eng. also develops technology and expertise supporting the society

Source: M. Shaw. Prospects for an eng. discipline of software. IEEE SW, 7(6): 15-24, 1990.

What is Engineering?

- Creating cost-effective solutions
  - Not just about solving problems, but doing so with economical use of all resources

by applying *scientific* knowledge

Let's look at Computer Science

- to build things
  - Emphasizing the solutions, which are usually tangible artifacts

- in the service of mankind
  - Not only serving the immediate customer, but eng. also develops technology and expertise supporting the society

Source: M. Shaw. Prospects for an eng. discipline of software. IEEE SW, 7(6): 15-24, 1990.

Building Blocks for Software

- Church-Turing Thesis: Any Turing-complete machine can compute anything that is computable.
- Implies that code running on any computer can (theoretically) fulfill any (computable) functional requirements.
- Problem: Within the space of what's computable, limitations come from our own limited capacities
  - What should be the requirements?
  - What languages & abstractions can we dream up?
  - What are our limitations and how can we compensate for them?
  - How can we act together in a coordinated way?

Source: J. Herbsleb. Building a theory of coordination: why and how. ACM SIGSOFT FSE, 2016

What is Engineering?

- Creating cost-effective solutions
  - Not just about solving problems, but doing so with economical use of all resources

by applying *scientific* knowledge

My view: "a human-centric perspective"

- to build things
  - Emphasizing the solutions, which are usually tangible artifacts

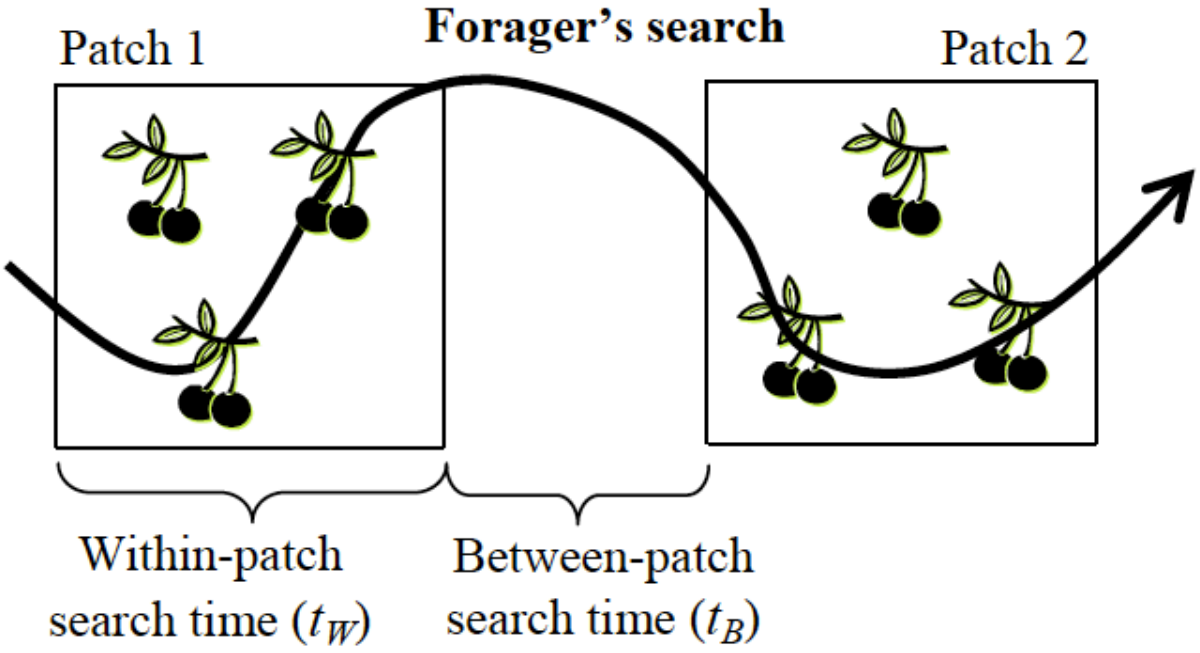
- in the service of mankind
  - Not only serving the immediate customer, but eng. also develops technology and expertise supporting the society

Source: M. Shaw. Prospects for an eng. discipline of software. IEEE SW, 7(6): 15-24, 1990.

Optimal Foraging Theory

[Stephens and Krebs, 1986]

- Basics
  - Animals adapt, among other reasons, to increase their rate of energy intake
  - Optimality here refers to the strategy that maximizes the gain per unit cost



$$R = \frac{G}{t_B + t_W}$$

Patchy Environment

Social Foraging Theory

[Giraldeau and Caraco, 2000]

⇨Hints are exchanged in group foraging

⇨e.g., individual performance affected by group size

$$R = \frac{G}{t_B + t_W}$$

$$R(n,H) = \frac{G/n}{t_I + t_B + t_W} = \frac{G/n}{\frac{1}{n\cdot\lambda(n)} + \frac{\lambda(H)}{n\cdot\lambda(n)} + \frac{\tau(n)}{n\cdot\lambda(n)}}$$

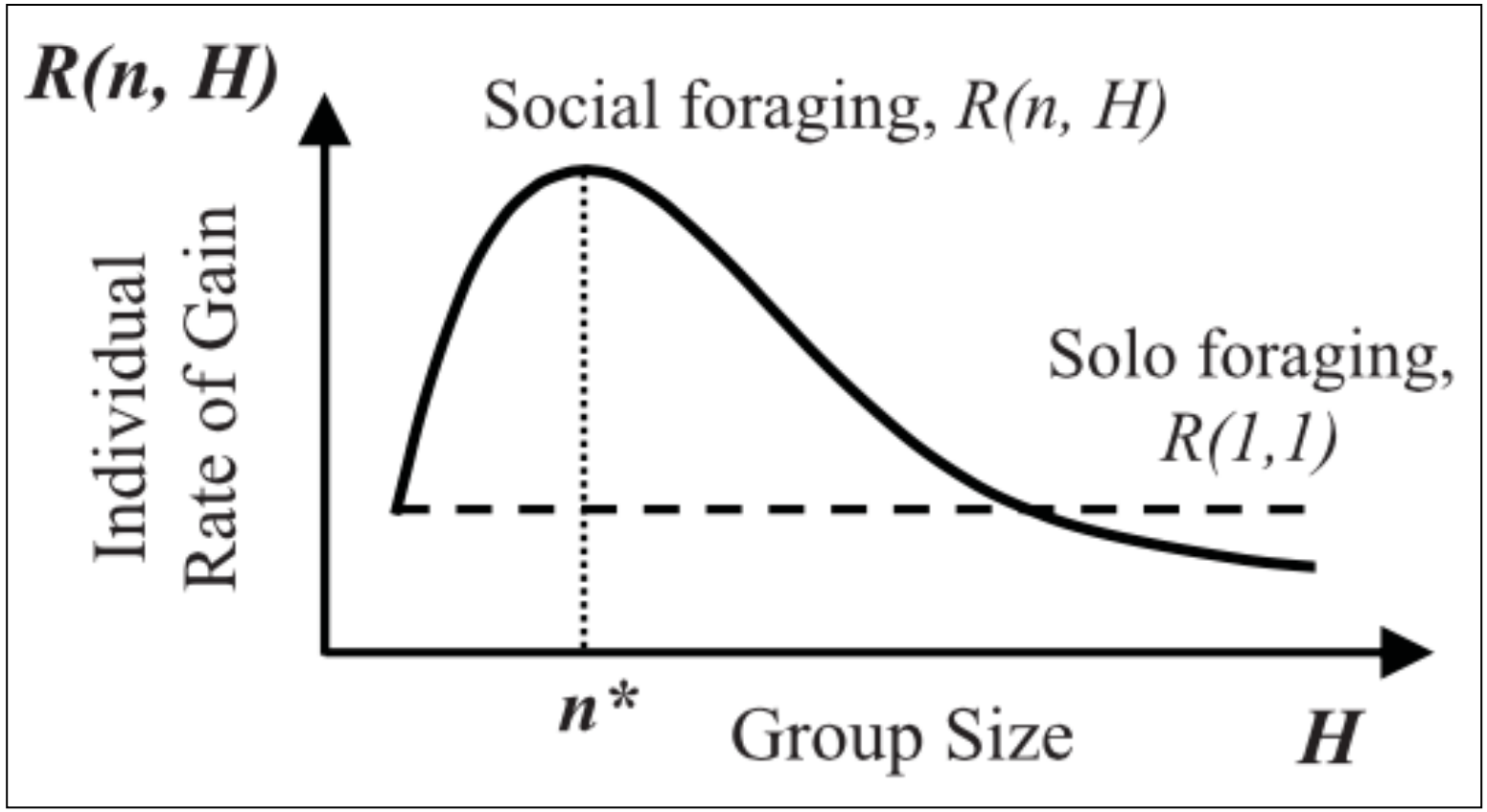
$$= \frac{\lambda(n)\cdot G}{1 + \lambda(H) + \tau(n)}.$$

Social Foraging Theory

[Giraldeau and Caraco, 2000]

⇨Hints are exchanged in group foraging

⇨e.g., individual performance affected by group size



Group Size for OSS Change Tasks

[Bhowmik *et al.*, 2016]

	Firefox	Mylyn
Domain	Web browsing	Task management
Source	mozilla.org/firefox	eclipse.org/mylyn
Programming Language	C/C++, JavaScript	Java
# of Source Code Files	1968 (C/C++)	2321
Analysis Begin Date	2004-07-06	2006-12-05
Analysis End Date	2011-06-20	2011-02-28
# of Unique Developer IDs	2569	149
# of Completed Tasks	2878	1898
# of Commits	18538	4908

Main Results

- H1: There's no difference between *n** (optimal group size) and *n* (actual group size).
  - Rejected.
- H2: The closer *n* is to *n**, the more productive the group members are.
  - Supported.
- H3: As the OSS project evolves, the group size becomes more optimal.
  - Rejected.
- H4: Developer's solo-social changes in evolution lead to productivity gain at the group level.
  - Initial evidence *(by looking at high-profile developers)*

Where's the Theory for Software Engineering?

Pontus Johnson, Mathias Ekstedt, and Ivar Jacobson

⇨Description

⇨Explanation

⇨Prediction

⇨Action

Where's the Theory for Software Engineering?

Pontus Johnson, Mathias Ekstedt, and Ivar Jacobson

⇨Description
⇨Explanation
⇨Prediction
⇨Action

Take-Away Messages

- ⇨ Software ≈ Peopleware
  - ⇨ Human as user & human as builder

- ⇨ Software engineering goes social
  - ⇨ How to build the software to enable its users to behave nicely?

- ⇨ Foraging theory
  - ⇨ What info. to hunt? How long to seek & handle it? What to exchange & what not? To whom and how widely? …

Studying the Social Aspects of Software Development

Nan Niu

http://homepages.uc.edu/~niunn

Thank You!