CSE 403
Software Engineering

Summer 2006

Pragmatic Programmer Tip: Care about Your Craft
Why spend your life developing software unless you care about doing it well?

Outline

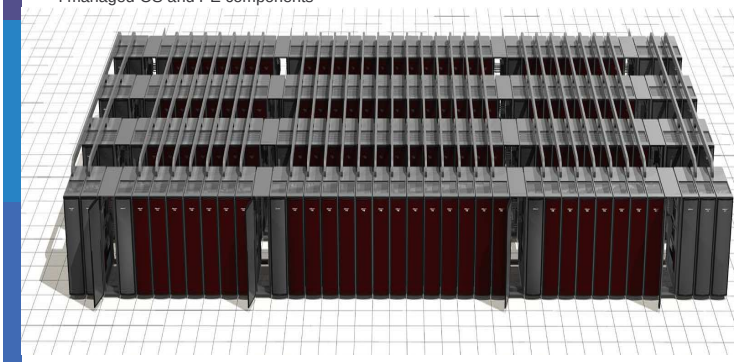
- My engineering background
- Brief software engineering discussion
- What it takes to be a great engineer

Who am I?

- Gail Alverson
- Computer Science Professor
- Teacher of CSE 403 in Spring 2006
- Senior Engineering Manager at Cray Inc.
- Your entertainer for the next 60 minutes!

Cray's Red Storm system

- Massively parallel processing supercomputer system used for analysis and stewardship of nuclear weapons - for Sandia National Lab $93M
- I managed OS and PE components



My SW Engineering roles

PhD University of Washington, CSE, 1990

Software Engineer – libraries and debugger

Project lead/Software engineer, libraries and tools

Manager, Programming Environments (PE)

Senior Manager, OS and PE components

Technical Manager for the VP of Engineering, 2006

TIME

My SW Engineering roles

PhD University of Washington, CSE, 1990

Software Engineer – libraries and debugger

Project lead/Software engineer, libraries and tools

Manager, Programming Environments (PE)

Senior Manager, OS and PE components

Technical Manager for the VP of Engineering, 2006

TIME

Can you think of a different SW career track?

What is software engineering?

Let's capture your ideas!

- Write on an index card, what you think "Software Engineering" is
- Please include your name, as we'll return it to you at the end of the quarter, so you can see if your thoughts have changed
- This activity will NOT be graded

What is software engineering?

Let's hear your ideas!

What is software engineering?

Let's hear your ideas!

- Here are some of mine:

Software engineering involves:

- *Processes* necessary to turn a concept into a robust deliverable that can *evolve* over time
- Working with *limited* time and resources
- Satisfying a *client*
- Managing *risk*
- *Teamwork* and communication

Here's another view:

- Software engineering is "social engineering"
  - Software is an intervention!
  - The client is asking you to change the way they do business
    - Means changing user's behavior
    - Means clients are an integral part of the system you are designing

Socha,06

What's the key point here?

What is a "software engineer"?

How does an engineer differ from a programmer?

What makes a great engineer?

Consider two related questions:

- What makes an effective software engineer?
- What's the difference between an adequate engineer and a great engineer?

Let's hear your ideas!

7 habits of highly successful engineers

Acknowledgements

- David Papworth (Intel Fellow)
  - Principle architect of the Intel processor designs that became Intel's Pentium Pro, Pentium II, III, and 4
- + Cray and 403 experiences!

Caring

- It's the final product that matters
- Not just your portion
- Not your career, not your focal ranking
- Not the indicators
- Pride in your work and pride in your peers

Caring examples

- Tera/Cray

Early startup days, engineers took a pay cut to enable the company to survive. 100% remained. They believed in the product. They believed in their coworkers.

Caring examples

- Tera/Cray

Early startup days, engineers took a pay cut to enable the company to survive. 100% remained. They believed in the product. They believed in their coworkers.

- 403

Railpad group (collaborative text editing) went beyond requirements to get product on Source Forge. Several students are continuing with the project, even now class is over.

Sharing

- For the product to succeed, all parts must succeed
- Often easier to solve new problems in one unit, than to fix existing problems in another
- When your neighbor asks for a cup of sugar, lend it to him

Anyone want to take a swag at interpreting these?

Sharing examples

- Cray

- Software workarounds to mitigate hardware bugs
- IO group runs a battery of test to track down a hyper-transport problem (owned by another group)
- 2005 "Call to Action" for Red Storm

Sharing examples

- Cray

- Software workarounds to mitigate hardware bugs
- IO group runs a battery of test to track down a hyper-transport problem (owned by another group)
- 2005 “Call to Action” for Red Storm

- 403

- Collaborative Imaging student created a TomCat tutorial to bring everyone up to speed
- Railpad student set up the development environment for several of the team

Responsibility

- Support your work
- 90% efforts don’t count at all
- Don’t make commitments, or otherwise say what management wants to hear, unless it’s true
- Exercise due care. Stakes are high

What happens if you aren’t responsible?

Responsibility examples

- Cray Example

Important to ship a product in Q40x. Engineers know this is a stretch, and provide a probability along with their commitment.

Responsibility examples

- Cray Example

Important to ship a product in Q40x. Engineers know this is a stretch, and provide a probability along with their commitment.

- 403 Example

Collaborative Art student in charge of database corrected every bug, added every feature, immediately on notice. Total support for her group, and they noticed.

Joy

- Engineering work should provide its own sense of satisfaction, beyond financial and peer recognition
- If you don’t genuinely enjoy solving that speed path, or fixing that software bug, this isn’t the right field for you

Joy examples

- Cray example

Red Storm team beaming as the Sandia machine ran its first job on thousands of processors

Joy examples

- Cray example

  Red Storm team beaming as the Sandia machine ran its first job on thousands of processors

- 403 example

  Railpad team beaming as they discussed their release to Source Forge

  DynamicGrid team beaming during their final demo, showing a grid app on all the machines in the lab

Growth

- Curiosity, versatility, and flexibility are as important as raw talent
- Never be afraid or unwilling to learn the other guy's job
- Don't always ask for help
- Don't always wait for mentoring
- You can do it on your own

Growth

- Curiosity, versatility, and flexibility are as important as raw talent
- Never be afraid or unwilling to learn the other guy's job
- Don't always ask for help
- Don't always wait for mentoring
- You can do it on your own

Pragmatic Programmer Tip: Invest regularly in your knowledge profile. Make learning a habit.

Growth examples

- Cray non-example

  New engineer struggles with a problem for days, before a senior engineer notices and gets him on the right track in minutes

Growth examples

- Cray non-example

  New engineer struggles with a problem for days, before a senior engineer notices and gets him on the right track in minutes

- 403 example

  Many students learned new tools as part of their project development: Ruby on Rails, Spring, TomCat, Bugzilla, ASP.net, …

Creativity

- Never give up on a problem
- There's almost always a way around it
- Resignation is self-defeating

Creativity

Can you think of a creativity example from your past?

- Never give up on a problem
- There's almost always a way around it
- Resignation is self-defeating

Creativity examples

- Cray Example

Linux bios init code wasn't working on new chip. Many brains on the problem didn't make progress. Engineer with bio background, thought of a genetic algorithm that could apply to situation. Yes!

Creativity examples

- Cray Example

Linux bios init code wasn't working on new chip. Many brains on the problem didn't make progress. Engineer with bio background, thought of a genetic algorithm that could apply to situation. Yes!

- 403 Example

TestFiles had a problem with iisql server and asp.net authentication. Talked with staff, students, finally consulted a database TA from a prior class, who recognized the problem and provided a solution. Yes!

Objectivity

- Having strong opinions is not "bad"
- Taking a firm position is often part of the job This is not arrogance
- Let the facts speak for themselves
- You can admit that you were wrong
- Resolve differences with objective data, not force of personality

Objectivity examples

- Cray Example

Two network designs being considered for a new machine, fat tree and torus. A lot of emotion behind each. Objectively gathered data and took to PRB to determine sound decision for company.

Objectivity examples

- Cray Example

Two network designs being considered for a new machine, fat tree and torus. A lot of emotion behind each. Objectively gathered data and took to PRB to determine sound decision for company.

- 403 Example

Collaborative Imaging stood their ground in their decision to use a file system instead of a database to store images

How do you develop these habits?

1. Acquire the requisite knowledge and skills
2. Be certain of your desire to be an engineer
3. Practice the actions listed previously until they become habits

How do you develop these habits?

1. Acquire the requisite knowledge and skills
2. Be certain of your desire to be an engineer
3. Practice the actions listed previously until they become habits

Is this simple?

What do you think it takes?

My advise to you for 403

- Communicate with your team! Over-communicate, even

- Revisit the “habits for highly effective engineers” weekly, to keep on track with becoming one yourself
- LIVE the lecture material - absorb it, embrace it, challenge it with your team and your project