this "SOFTWARE" stuff

PETE GOODLIFFE
Who is Pete?

Pete Goodliffe
pete@cthree.org
http://www.goodliffe.net
The pension plan #1
The pension plan #1
The pension plan #2

Alphabetti Spaghetti

Alphabeticus Spagheticus
The pension plan #2

Custard
*Dessertus Yellowicus*
The pension plan #2
The observation
The observation

THE

WRONG THING

THE

WRONG WAY
The goal

THE

RIGHT THING

THE

RIGHT WAY
Ask yourself . . .

Do I...
Ask yourself . . .

Do I . . .

...want to improve as a programmer?

Do I want to write the right thing in the right way?
This software stuff
This software stuff

SCIENCE

ART

GAME

CHORE

SPORT

OTHER

* CRAFT
* ENGINEERING
* RELIGION
* CROCHET
* BOXING
* MUSICAL COMPOSITION
The ideal programmer
The ideal programmer
The Artist

Software is... an art
Some programs are elegant, some are exquisite, some are sparkling.

My claim is that is it possible to write grand programs, noble programs, truly magnificent ones!
Software is...an art

**CREATIVE**
- INDIVIDUAL
- PERSONAL
- IMAGINATIVE
- SKILLFUL
- PRECISE

**AESTHETIC**
- ELEGANCE
- BALANCE
- INTERPRETATION
- IDIOMATIC & CULTURAL
- FORM & FUNCTION
- INTERFACES

**MECHANISM**
- OUR MEDIUM & TOOLS
- PROCESSES & TECHNIQUES
- BENEFACCTORS/COMMISSION

**TEAM-BASED**
- ORCHESTRA CONDUCTOR
- COMPOSER
- MOVIE DIRECTOR
- ARCHITECT
Michelangelo Buonarroti

“"I looked into the stone and saw him [David] there, and just chipped away everything else.""
Do I...

...consider the creative aspects of software development, or treat it as a mechanistic logical activity?
Ask yourself...

Should I...

...develop a keener sense of elegance and aesthetics in code?

Should I look beyond what’s functional and solves the immediate problem?
Do I...

...think that my idea of 'beautiful' code is the One True Opinion?

Should I consider artistry as a team pursuit?
Software is... a science
Any intelligent fool can make things bigger, more complex, and more violent.

It takes a touch of genius - and a lot of courage - to move in the opposite direction.
I am enough of an artist to draw freely upon my imagination. Imagination is more important than knowledge. Knowledge is limited. Imagination encircles the world.
Software is . . . a science

RIGOUROUS

* * *

ACCURATE
PROVEN
MEASURED
TESTED AND VERIFIED

SYSTEMATIC

* * *

INTELLECTUAL
LOGICAL
RATIONAL
BRINGS ORDER/UNDERSTANDING

INSIGHTFUL

* *

FORMING HYPOTHESES
APPLYING SCIENTIFIC METHOD

e=mc^2
Ask yourself...

Is my...

...software always totally correct and completely accurate?

How do I prove this? How can I make this explicit, now and in the future?
Do I...

...strive to bring order out of chaos?

Do I collapse complexity in my code until there are a few, small, unified parts?
Ask yourself...

Do I...

...approach problems methodically and thoughtfully, or do I rush headlong into them in an unstructured way?
Software is a sport.
Software is... a sport

TEAMWORK
* REQUIRES MANY SKILLS...

RULES
* RULES OF THE GAME
* TEAM CULTURE

DISCIPLINE
* TRAINING
* COMMITMENT
* HARD WORK
Teamwork characteristics

- PATIENCE
- COMMUNICATION
- SHARING
- PRACTICE
- COMMON GOAL & VISION
- RESPECT
- RESPONSIBILITY
- TRUST
- FRIENDSHIP
- FOOD
- FUN
- TRUST
- FUn
Ask yourself...

Do I...

...have all of these skills?
Do I work well in a team, or could I improve in some areas?
Software is... a sport

TEAMWORK
* REQUIRES MANY SKILLS...

DISCIPLINE
* TRAINING
* COMMITMENT
* HARD WORK

RULES
* RULES OF THE GAME
* TEAM CULTURE
Ask yourself...

Am I...

...still learning about software development?

Do I learn from others, and am I perfecting my team moves?
Do I...

...invest enough effort in my own development? Am I continually in training?

Am I tired and muddy?
Software is... child’s play
Do I...

...write the simplest code possible?

Or do I type what comes to mind, and not think about commonality, refactoring, or code design?
Am I...
...still learning?
What can I learn about?
What do I need to learn about?
Software is... a chore

The Janitor

*Software is... a chore*
CLEANING UP

* SPOTTING AND ADDRESSING PROBLEMS
* NOT LEAVING IT TO SOMEONE ELSE
* RESPONSIBILITY

BACKGROUND

* WORKING OUT OF THE LIMELIGHT
* NO RECOGNITION FOR HEROIC EFFORTS
* A "SUPPORT" ROLE

MAINTENANCE

* REMOVING CODE
* REFAC'TOR CODE
* TIDYING UP
* DONE OFTEN/WHEN NECESSARY
Am I...

...happy to do code 'chores'?

Do I only want the glamorous work?
Do I...

...take responsibility for messy code and clean it up?
The ideal programmer
One final question

What are you going to do now to help you to write...

The right thing the right way
The book... buy it now!

“Its **useful and fun** and it’ll make you a better programmer.” Jez Higgins “A **goldmine of information** that every professional software developer should be aware of.” Tim Penhey “A **terrific resource** for developers wanting to learn or teach good coding practices ... deserves a place on the bookshelf.” Frazzled Dad blog “A **unique and practical** guide to being a professional programmer in the modern workplace.” Andrew Burrows “**Readable, engaging, and even funny** ... It's the book I wish I'd had when I started work as a programmer.” Steve Love “A 'must read' for any programmer who wants to be a better programmer” Linux Tutorial “This is **exactly** the kind of book you should give raw recruits.” Jon Jagger
this "SOFTWARE" stuff

Pete Goodliffe