Bloomberg

C++: Engineers Wanted, Programmers not so Much

David Sankel

C++Now

May 7th 2019

Why do so many successful projects have such bad code?

Why hasn't functional programming taken off?

What makes a good API?

This is a philosophical talk

Why do so many programmers do non-programming tasks?

Why is agile "by the book" so rare?

Why is C++ still so popular?

Why do so many programmers avoid the STL?

Bloomberg


RON
FRANCISCO
MONTERO
RON
MONTERO
RESERVA


FRANCISCO MONTERO
RON MONTERO
RESERVA

Antonio, what do you do?

Jane programmer, what do you do?

I write C++ and Python. I have experience with Qt and think Boost is amazing.

Elena Software Engineer, what do you do?

I'm responsible for a message router that handles, on average, a billion requests per second under continuous load and is the backbone of Bloomberg's financial engine.


Obviously, aesthetics are a minor factor in rocket design.
- Elon Musk

Antonio’s Role

- Operations
- Renovations
- Integrations
- Design


ALARMAS
MODO CONCENTRACION AUTOMATICO
DENSIDAD ÁREA 62
MODO CONCENTRACION AUTOMATICO
ÁREA 62
MODO CONCENTRACION AUTOMATICO
CAUDALES ÁREA 62
MODO CONCENTRACION AUTOMATICO
TERBEL
ENTERADO ALARMA
11:31:06
25/04/2019

Antonio’s interesting integrations

- **Move a plant**
- **Restore a sugar cane factory in the Dominican Republic**

Engineering Priorities

- Long term demonstrated success vs. new tech
- Brand reputation matters
- Prefer used to new
- Extensive sharing between companies

clang-format

Bloomberg

Innovation for an engineer vs. innovation for a programmer

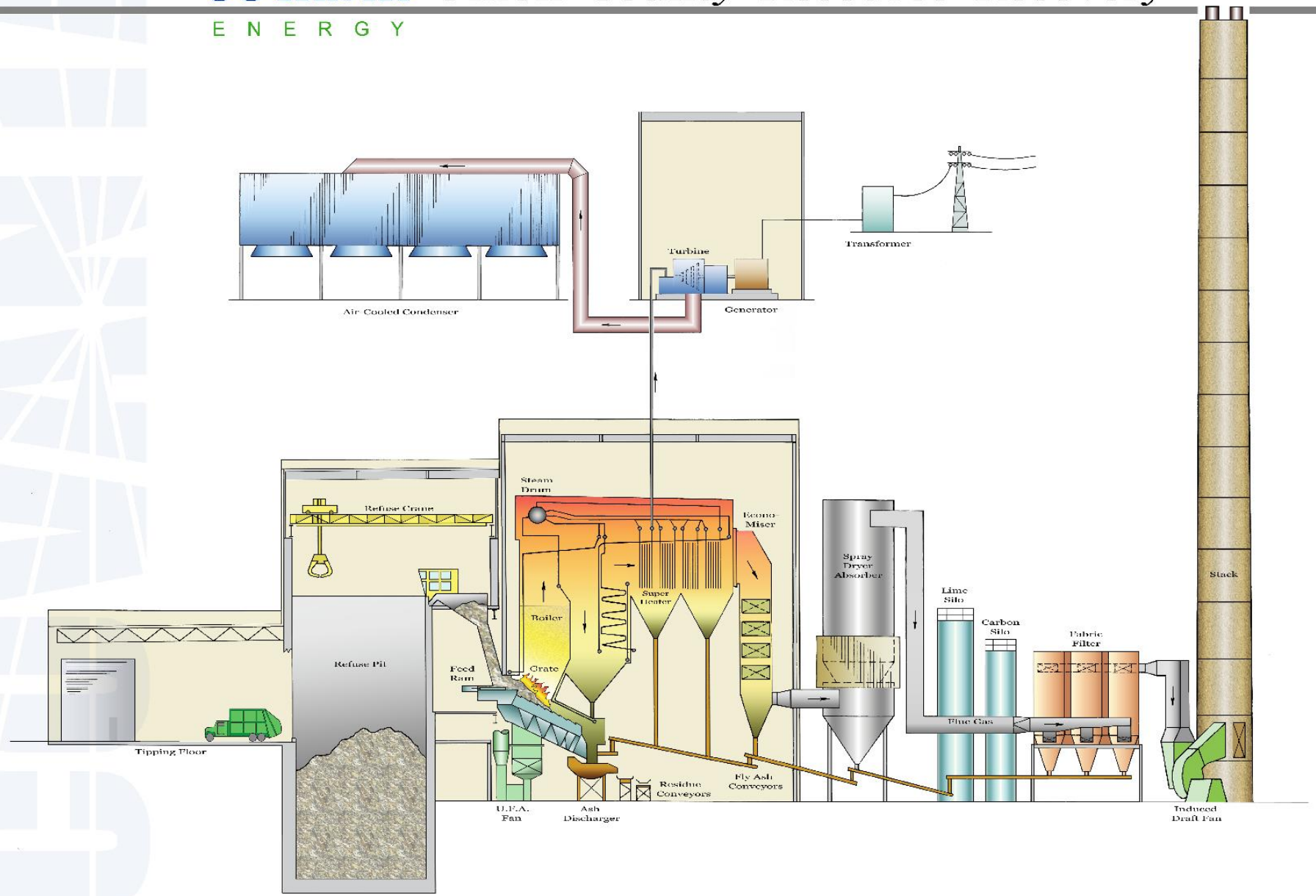
Covanta Union, NJ

- Processes 1,500 tons of solid waste per day
- Generates up to 42 megawatts
- Powers 30,000 homes and businesses


Clean Power
COVANTA
Covanta Union County Recent Emissions Performance

COVANTA ENERGY

Union County Resource Recovery



Service Level Objectives (SLOs)

- 95% uptime for each of the three boilers
- 99% uptime for the turbine
- Based on historical data and is a bit out of reach

Do you have sporadic test failures with your CI system?

- Do you have a SLO?
- Do you measure your service level indicator?

Turbine overspeed test

- Verifies that a greater load can be handled
- A scheduled risk to prevent unscheduled risks

Ingonel project

- **Tube coating that prevents failures.**
- **Incremental deployment:**
  - Start at critical sites
  - Apply slowly over time due to cost constraints

Periodic maintenance

- **2 x per year each boiler is taken down**
- **Once every 4-5 years the turbine is taken down**
  - Revisit everything
  - Take apart and put back together again

Renovations sometimes cause outages

- **Turbine improvements can cause increased vibrations.**
- **When this happens, put a hold on the periodic testing.**

Key idea, spread risk over time!

Spread risk in Software Engineering

- Continuous deployment
- Incremental changes

Migration Failure Checklist

- The “2.0” migration
- Opt-in migration
- One migration tied to another migration
- Deploy all at once

Many are stubborn in pursuit of the path they have chosen, few in pursuit of the goal.

- Nietzsche

Software Fitness Theory

- **Software is a means to an end**
  - End is not in the temporal or fixed sense
- **Fitness is observed by measured proximity to the end**

Convictions are more dangerous foes of truth than lies.

- Nietzsche

Ideologies

- Ideologies make for easy dopamine hits
- The “fan boy” measures software quality by proximity to the ideology
- Proximity to an ideology is unlikely to be a good measure of software fitness
- Ideologies, by their very nature, are oversimplifications and lead to absurdities

Ideology: Don’t Repeat Yourself (DRY)

```
int main(int argc, char** argv) {
    // …
}
```

Ideology: Don’t Repeat Yourself (DRY)

```
#include <mainlib>

MAINLIB_MAIN {
    // …
}
```

Ideology: Adam Wiggens's 12 Factors

1. Once codebase tracked in revision control, many deploys
2. Explicitly declare and isolate dependencies
3. Store config in the environment
4. Treat backing services as attached resources
5. Strictly separate build and run stages
6. Execute the app as one or more stateless processes
7. …

If you don't rely on feedback, then it isn't decent engineering

- Bjarne Stroustrup

Fitness is not just social

Limitations of correctness ideology

```
// Return the sum of the specified ‘lhs’ and ‘rhs’
int sum(int lhs, int rhs);
```

Limitations of correctness ideology

```
// Sort the specified ‘list’
void sort(std::vector<int> * list);
```

Limitations of correctness ideology

```
// Sort the specified ‘list’ in O(n*log n) time.
void sort(std::vector<int> * list);
```

- Runtime fitness
- Security fitness

How can we measure security fitness?

A team of security experts constantly try to hack your application and are unable to do so.

A good way to achieve fitness is to put your candidate in a hostile environment.

Some hostile environments

- Code review
- Manufactured break-in attempts
- UI feedback
- API feedback
- Stress testing

Software Fitness:
Unification of Software Engineering Disciplines

- Code reviews
- Stress testing
- Incremental improvements
- Staged deployment
- Security teams
- (others to come)

Software Fitness:
Answers many open questions

- Why do so many successful projects have such bad code?
- Why hasn’t functional programming taken off?
- What makes a good API?
- Why is agile “by the book” so rare?
- Why do so many programmers avoid the STL?

Software Fitness:
Understanding Machine Learning

On the morning of March 23, 2018 Walter Huang's Tesla allegedly took him out of his lane, pointed him at a fixed concrete barrier, then accelerated.

Software Fitness:
Artificial Intelligence

C++ Success

On April 8th, 2019 ZDNet reported that C++ rose to 3rd place in Tiobe's programming language popularity index for 2019, displacing Python.

Generalized fitness theory

- Music
- Ethics
- ?

Questions?

Bloomberg