

GEM
the Entertainment Maker

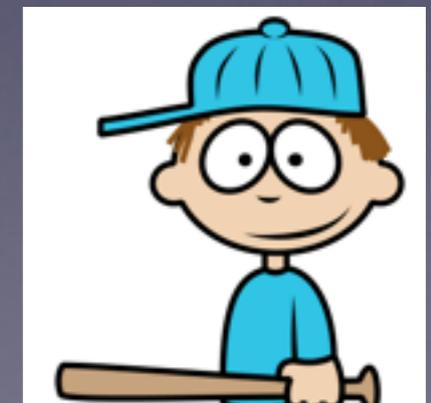


The GEM Team



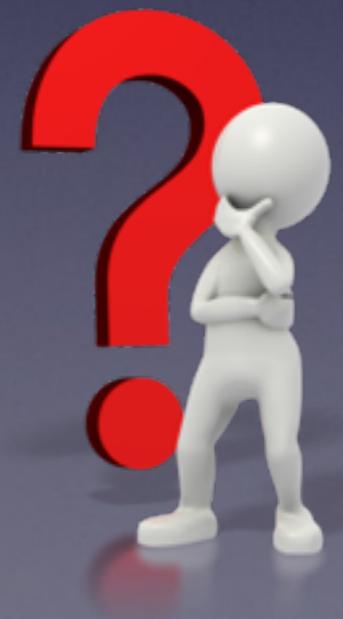
The GEM Team

- **Project Manager:** Tianlong Li
- **Language Guru:** Yuxuan Wang
- **System Architect:** Lixin Yao
- **System Integrator:** Xinyue Li
- **System Tester:** Hong Guo

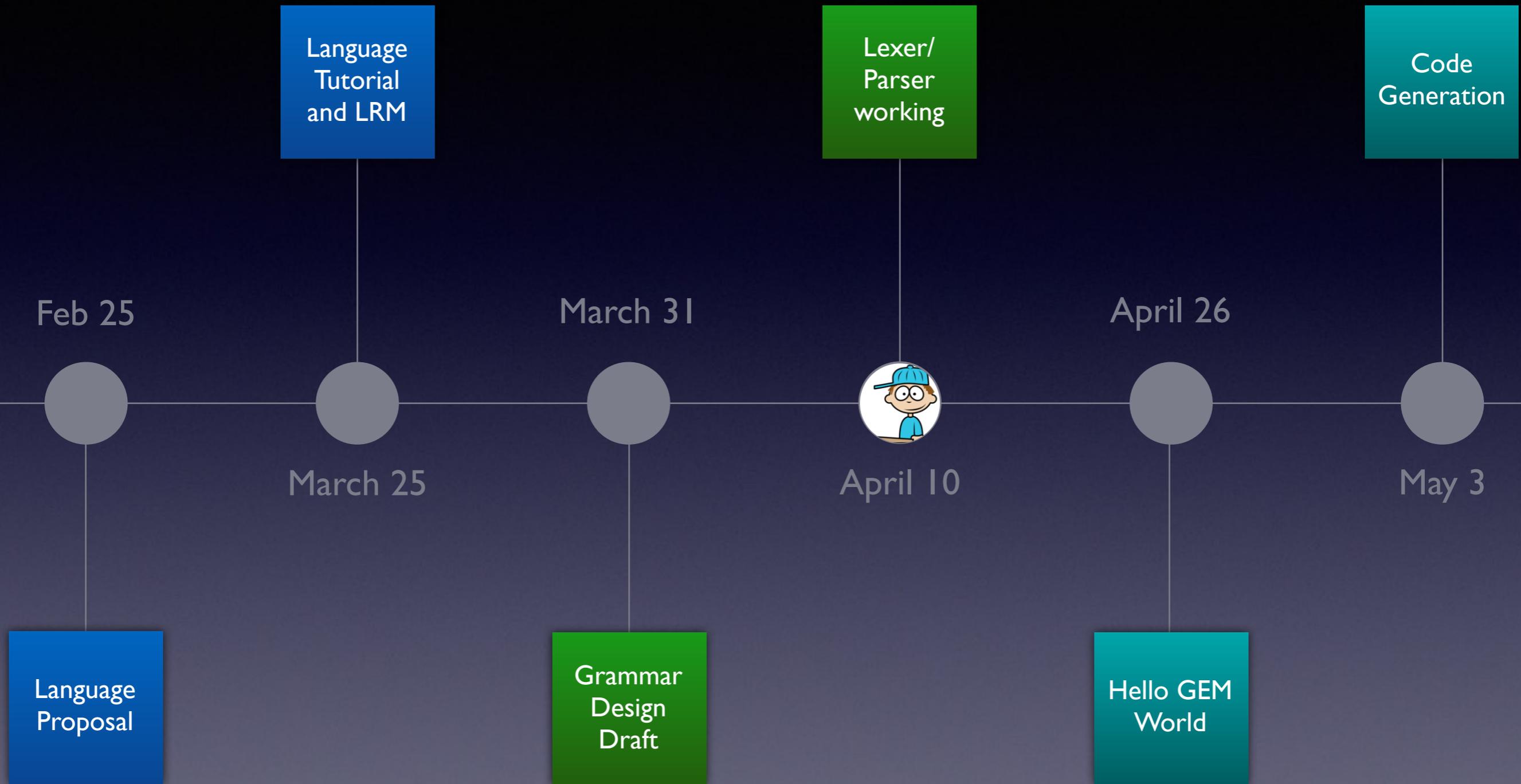


What is GEM?

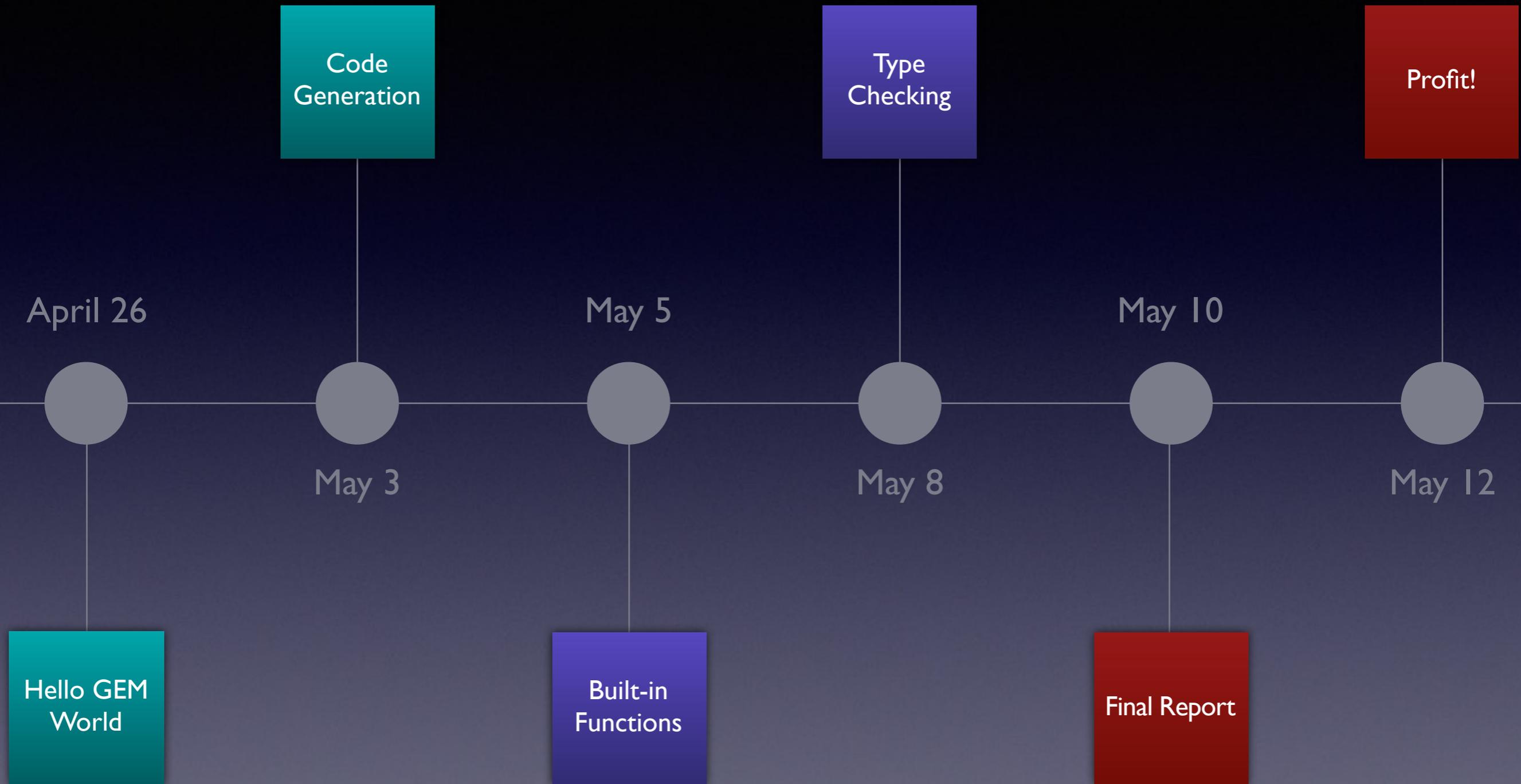
- Actually there were two ideas...and one won.
- Game designing and programming at one time.
- Name coined by Language Guru.
- Statically typed and game-oriented.



Timeline



Timeline

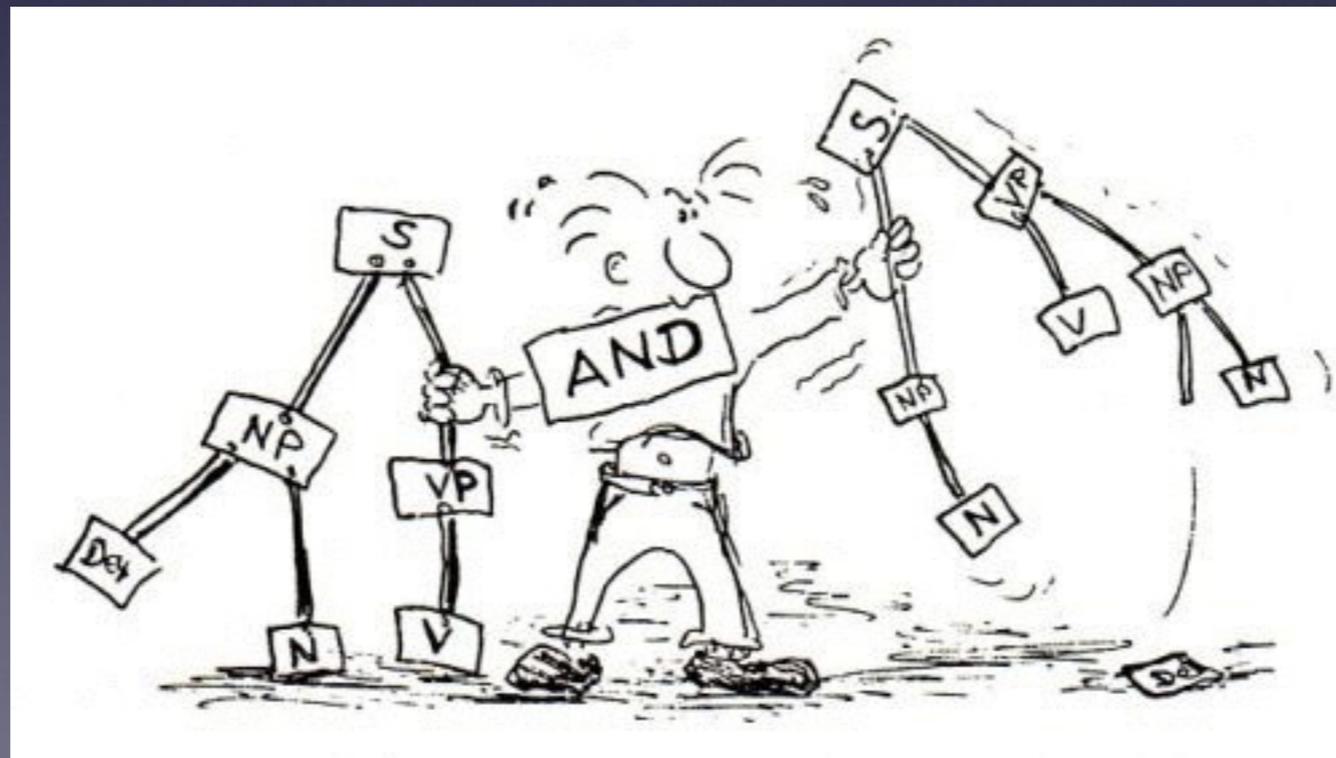


Tools for productivity and management



Syntactic Constructs

Yuxuan Wang - Language Guru



Sample code

```
/*
   sample demo program, will end automatically
*/
Skill heal = new Skill("Review", 0, 3, 0, 0, 0);
Skill[] skills = {heal};
Unit hero = new Unit("hero", 10, 10, 10, 1, skills);
Unit monster = new Unit("big boss", 15, 10, 19, 2, skills);
Event event;
Battle b = new Battle("How dare you come!", monster);

void main (String[] args) {
    Event[] end_event = {null};
    event = new Event("3", "fight!!!", end_event) {
        //our hero begin the battle
        if (hero trigger b)
            print "you win";
        else
            print "you lose!!!";
        next 0; //will terminate at end_event[0], which is null
    };
    run event;
}
```

Event

```
Event event;
```

```
Event[] end_event = {null};  
event = new Event("3", "fight!!!", end_event) {  
    //our hero begin the battle  
    if (hero trigger b)  
        print "you win";  
    else  
        print "you lose!!!";  
    next 0; //will terminate at end_event[0], which is null  
};  
run event;
```

String eventId, String display, Event[] options

Skill

```
Skill heal = new Skill("Review", 0, 3, 0, 0, 0);  
Skill[] skills = {heal};
```

String skillName, double healthMod,

int chiMod, double aMod, double dMode, int cost

Unit

```
Unit hero = new Unit("hero", 10, 10, 10, 1, skills);  
Unit monster = new Unit("big boss", 15, 10, 19, 2, skills);
```

String unitName, double attack,

double defend, double health, int chi, Skill[] skillSet

Unit

```
Battle b = new Battle("How dare you come!", monster);
```

```
if (hero trigger b)  
    print "you win";  
else  
    print "you lose!!!";
```

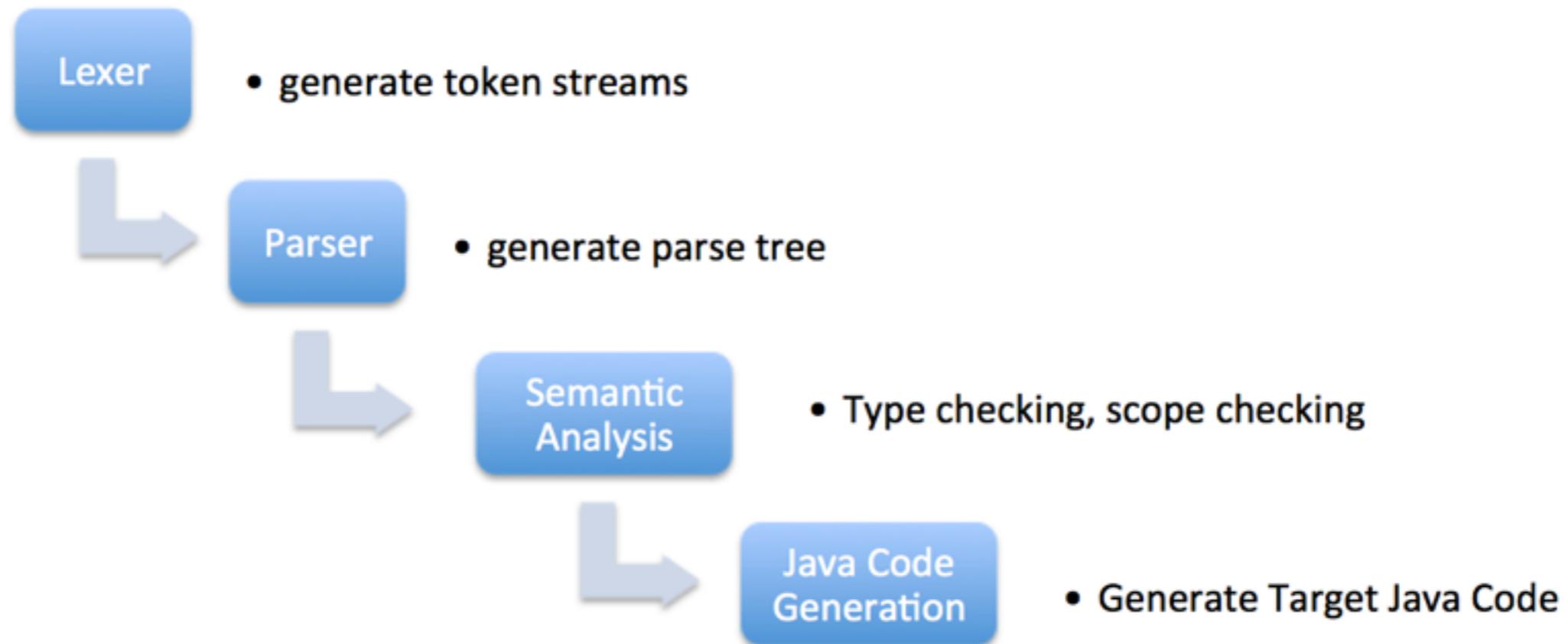
String display, Unit boss

Translator Architecture

Lixin Yao - System Architect



Translator Architecture



GENERATION
JAVA CODE

- Generate Target Java Code

Sample Code

```
1  /*
2  |   sample demo program, will end automatically
3  */
4  Skill[] skills = {}; //empty array, no skills
5  Unit hero = new Unit("hero", 10, 10, 10, 1, skills);
6  Unit monster = new Unit("big boss", 15, 10, 19, 2, skills);
7  Event event;
8  Battle b = new Battle("How dare you come!", monster);
9
10 void main (String[] args) {
11     Event[] end_event = {null};
12     event = new Event("3", "fight!!!", end_event) {
13         //our hero begin the battle
14         if (hero trigger b)
15             print "you win";
16         else
17             print "you lose!!!";
18         next 0; //will terminate at end_event[0], which is null
19     };
20     run event;
21 }
22
```

Generated Target Code

```
1 import java.util.*;
2 import buildinClass.*;
3 interface Plot {
4     void run();
5 }
6 public class Main {
7     public static Scanner scanner = new Scanner(System.in);
8     public static Map<String, Plot> plotMap = new HashMap<String, Plot>();
9     public static Skill[] skills = { };
10    public static Unit hero = new Unit("hero", 10, 10, 10, 1, skills);
11    public static Unit monster = new Unit("big boss", 15, 10, 19, 2, skills);
12    public static Event event;
13    public static Battle b = new Battle("How dare you come!", monster);
14    public static void main(String[] args) {
15        Event[] end_event = { null };
16        event = new Event("3", "fight!!!", end_event);
17        plotMap.put("3", new Plot() {
18            public void run() {
19                System.out.println("fight!!!");
20                if(b.trigger(hero))System.out.println("you win");
21                else
22                    System.out.println("you lose!!!");
23                if (end_event[0] != null)
24                    plotMap.get(end_event[0].id).run();
25            };
26        });
27        plotMap.get(event.id).run();
28    }
29 }
30
```

Type Checking Example

variable undefined
variable already defined
invalid operation
return type mismatch
parameters mismatch
illegal name usage
invalid unary operation
method undefined
return value missing
invalid index
continue, break error
run, trigger error
array initialization error
event initialization error

***Catch possible bugs
at compile time!!!***



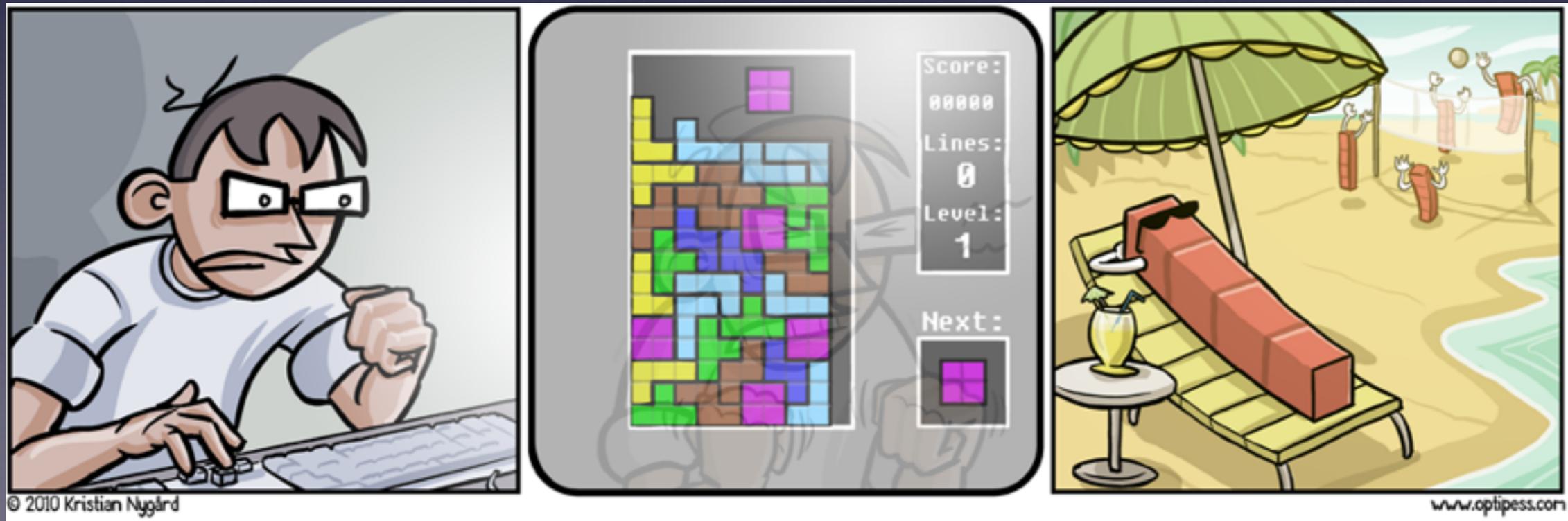
Sample Error Code

```
1  /*
2     sample demo program to show type checking
3  */
4  int add(int a){
5     return a+1;
6  }
7  int foo(){
8
9  }
10 int q = 1;
11 void main (String[] args) {
12     int a = "a";
13     int b = c;
14     int q = 2;
15     d = add("b");
16 }
```

```
Staniel@dyn-129-236-216-56:~/plt-code/PLT-GEM/Compiler$ ./gemc SystemTests/test
Type/demo2.gem
GEM Error on line 7 at position 0: No return statement for type int.
GEM Error on line 12 at position 5: Invalid operation on int and String.
GEM Error on line 13 at position 9: c is not defined.
GEM Error on line 14 at position 5: Duplicate definition of q.
GEM Error on line 15 at position 1: d is not defined.
GEM Error on line 15 at position 5: Parameters mismatchint.
```

Runtime Environment

Xinyue Li - System Integrator



Runtime Environment



- Mac OS X, Linux, Windows
- Java Runtime Environment 1.8
- GEM 1.0
- Shell Compiler Script

How to run?

```
1 ./gemc presentation.gem
2 ./gem |
```

gemc

```
1 java -jar GemC.jar $1 > Main.java
2
3 ./Artistic/bin/astyle -n -q Main.java
4 |
```

gem

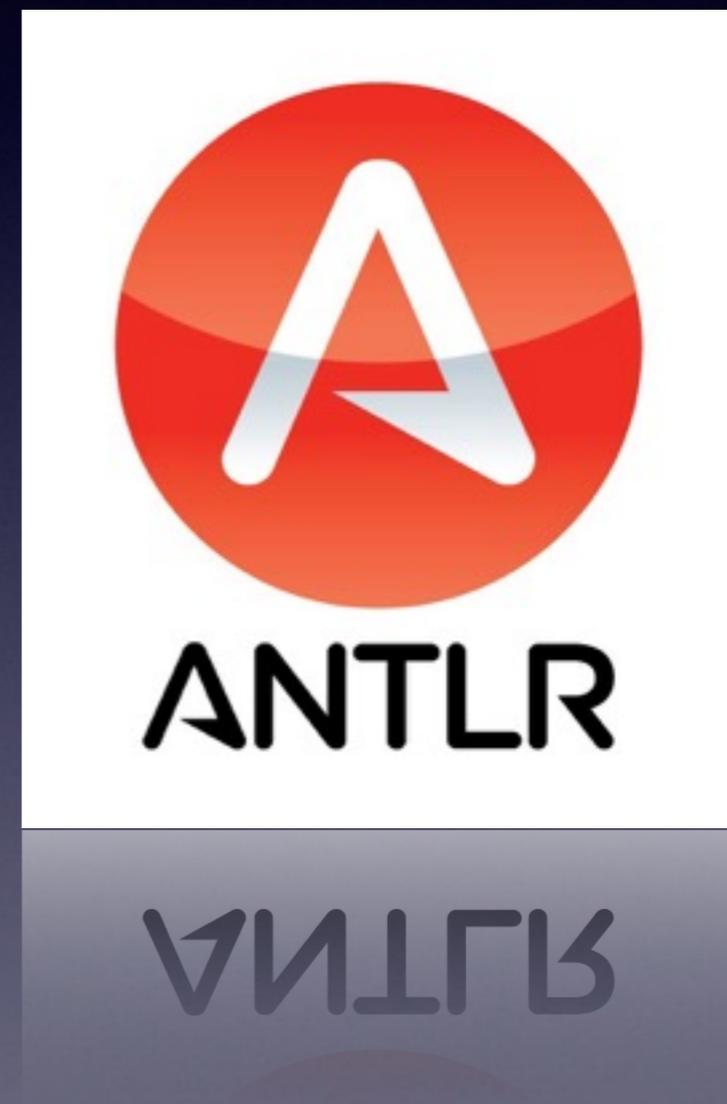
```
1 javac -cp GemC.jar Main.java
2
3 java -cp GemC.jar Main
4
```

Software Development Environment



Compiler-generator tool

- GEM.g4
- GEMLexer.java
- GEMParser.java
- GEMExtendVisitor.java
- GEMTypeCheckVisitor.java



Testing

Hong Guo - System Tester

How to raise bugs in 3 simple steps



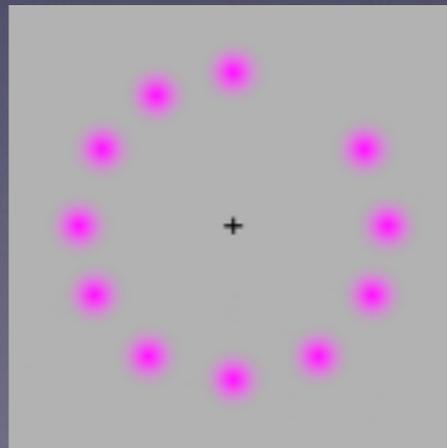
Testing Frameworks

We used two testing systems in parallel to test our programs:

1. JUnit tests: “micro test”, tests individual module or function.
2. System tests: “macro test”, compile all the tests automatically; compare output of tests to the expected output.



JUnit
Testing Framework



Conclusion

Tianlong Li - Project Manager



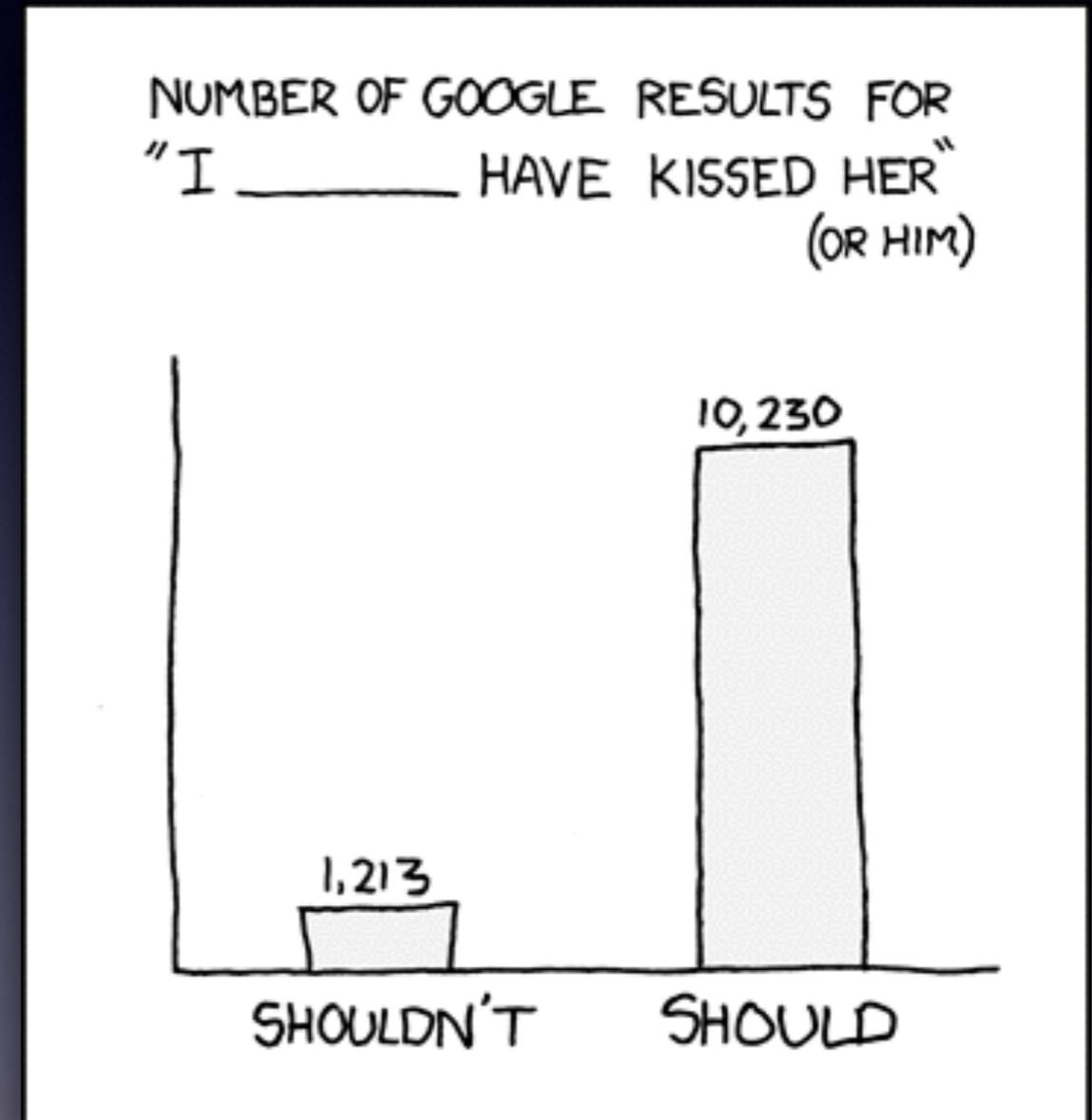
What worked well

- Tasks that everyone loves working with most interests.
- Parallel development.
- Fully Git-ified workflow.
- Very productive meeting and unhealthy happy hours.



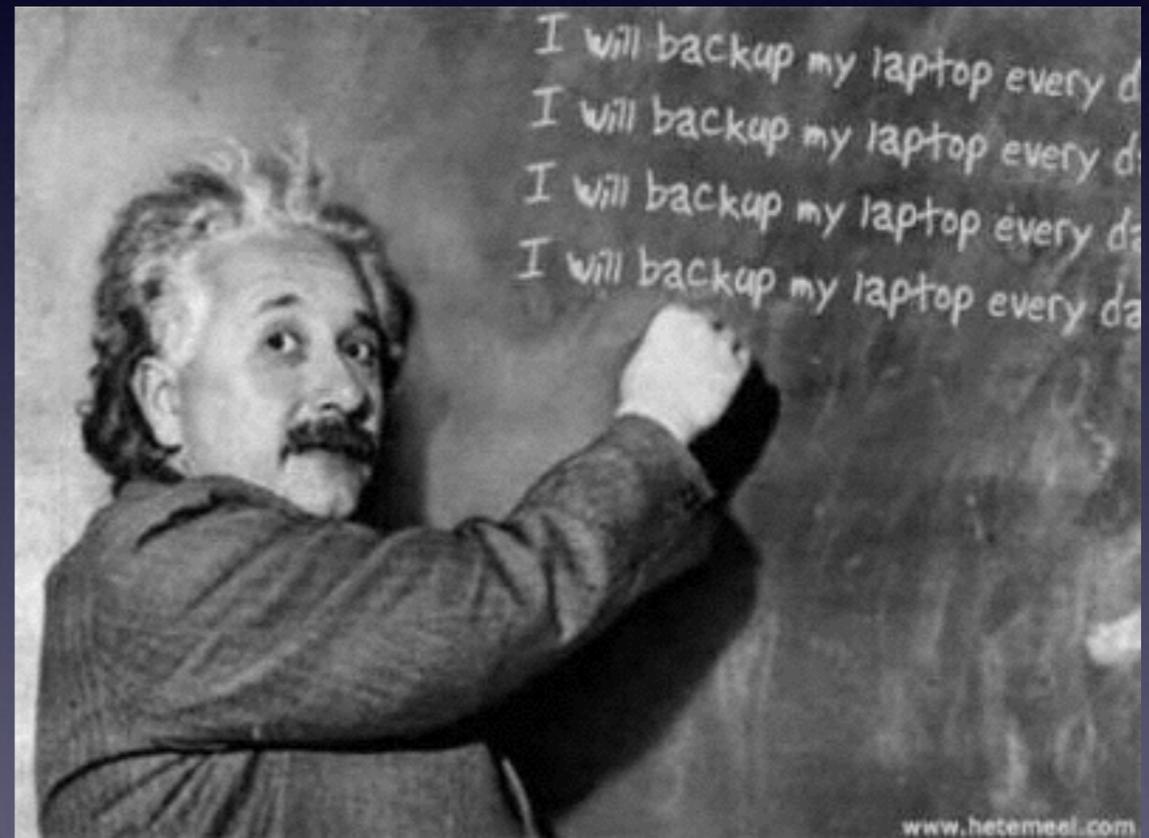
We should have...

- “Come talk to me” - **Alfred Aho the Canadian**
- Agile: remember Hello World?
- Meet ~~grow fat together~~ more often.



Lesson Learned

- Making a language is not so dreadful as it seemed.
- But testing takes much effort!
- Good idea makes you LOL and people nearby suffer.
- There's lesson to learn in a hard way, and totally worth it.



Why GEM for me?

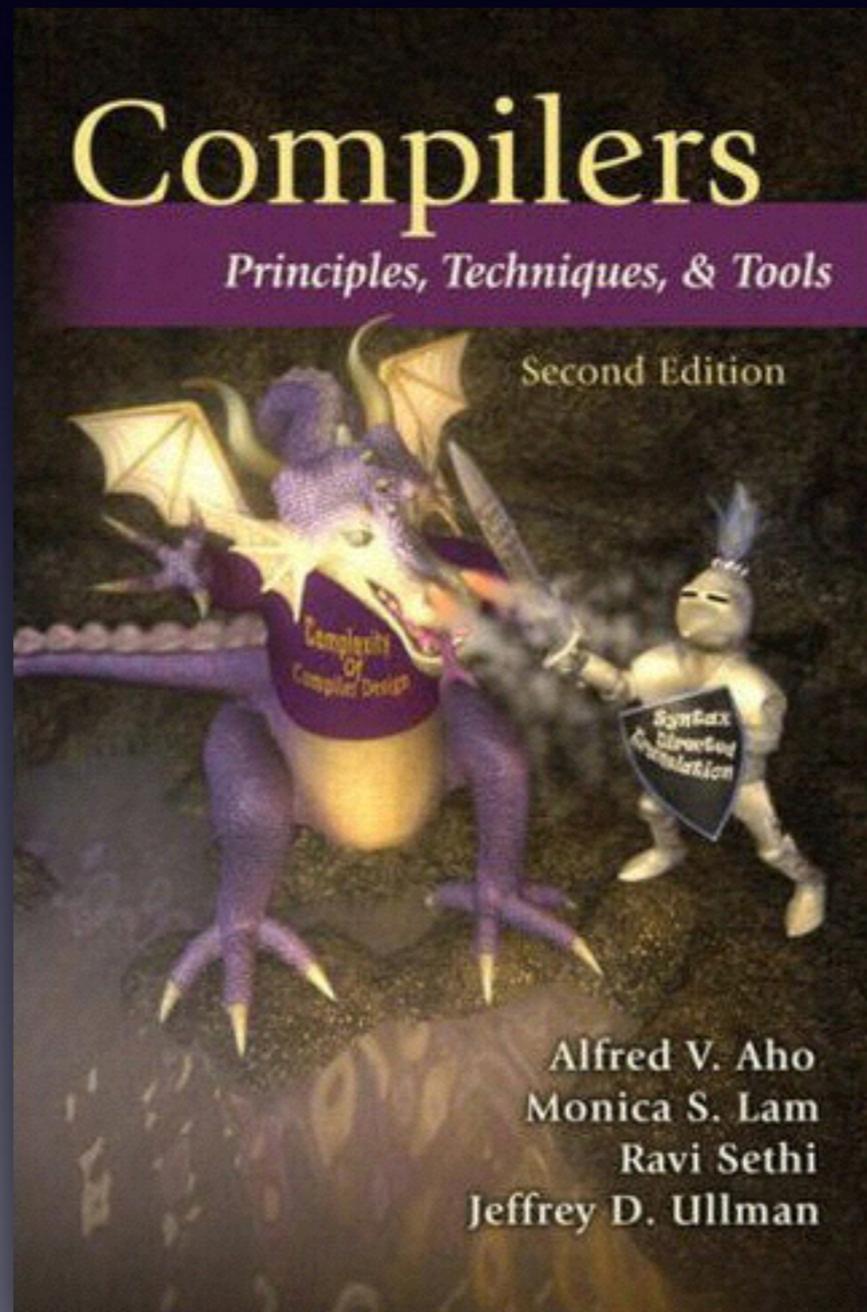
“Cheap but effective!”

–Anonymous System Architect



Demo

Adventure of PLTer





Thank you!

