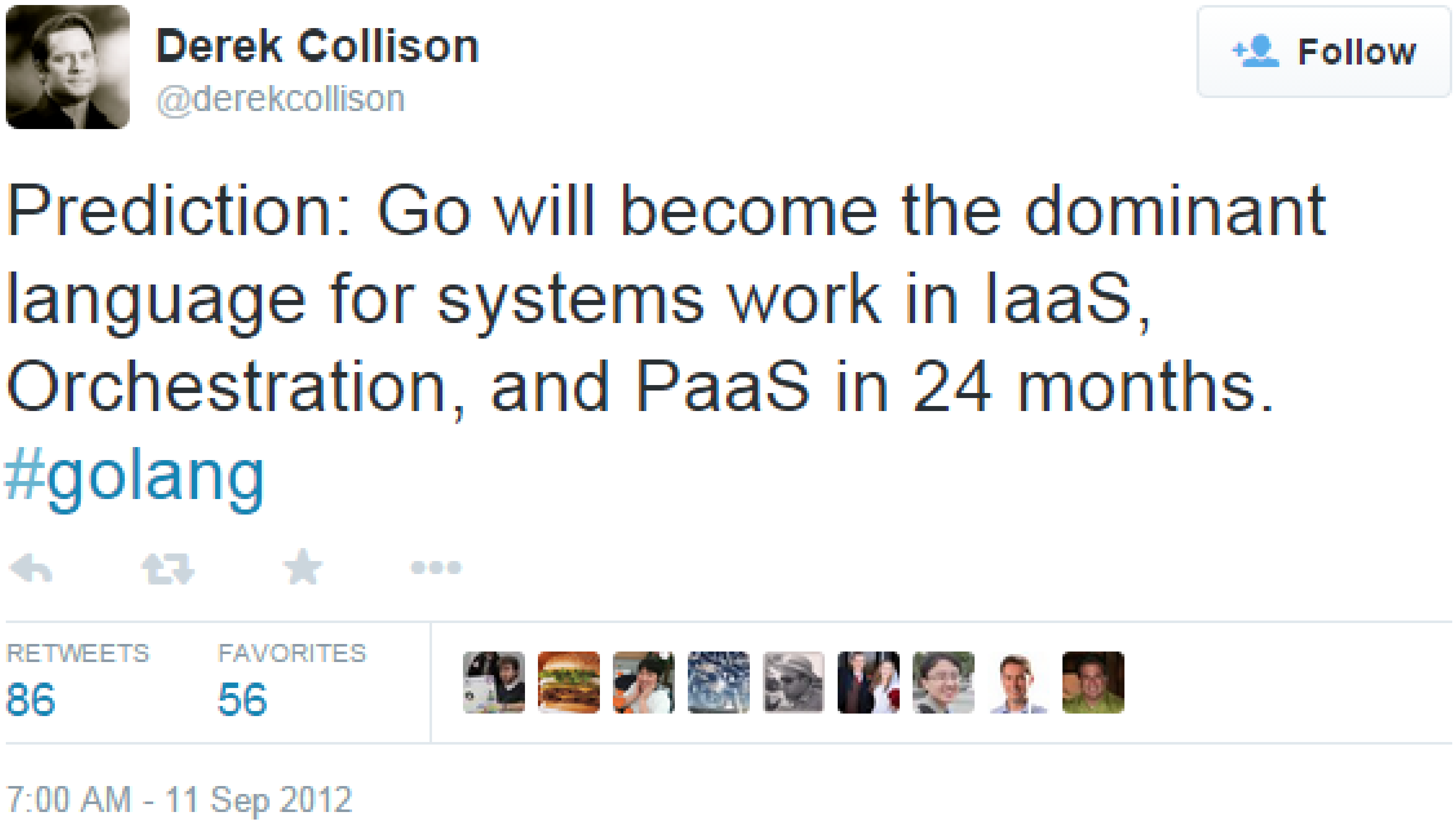
Introducing Go

A Powerful New Software Development Language

15 July 2015

Cem Ezberci
Principal Architect, GE Software

Fred Schults
Principal Architect, GE Software



Derek Collison
@derekcollison

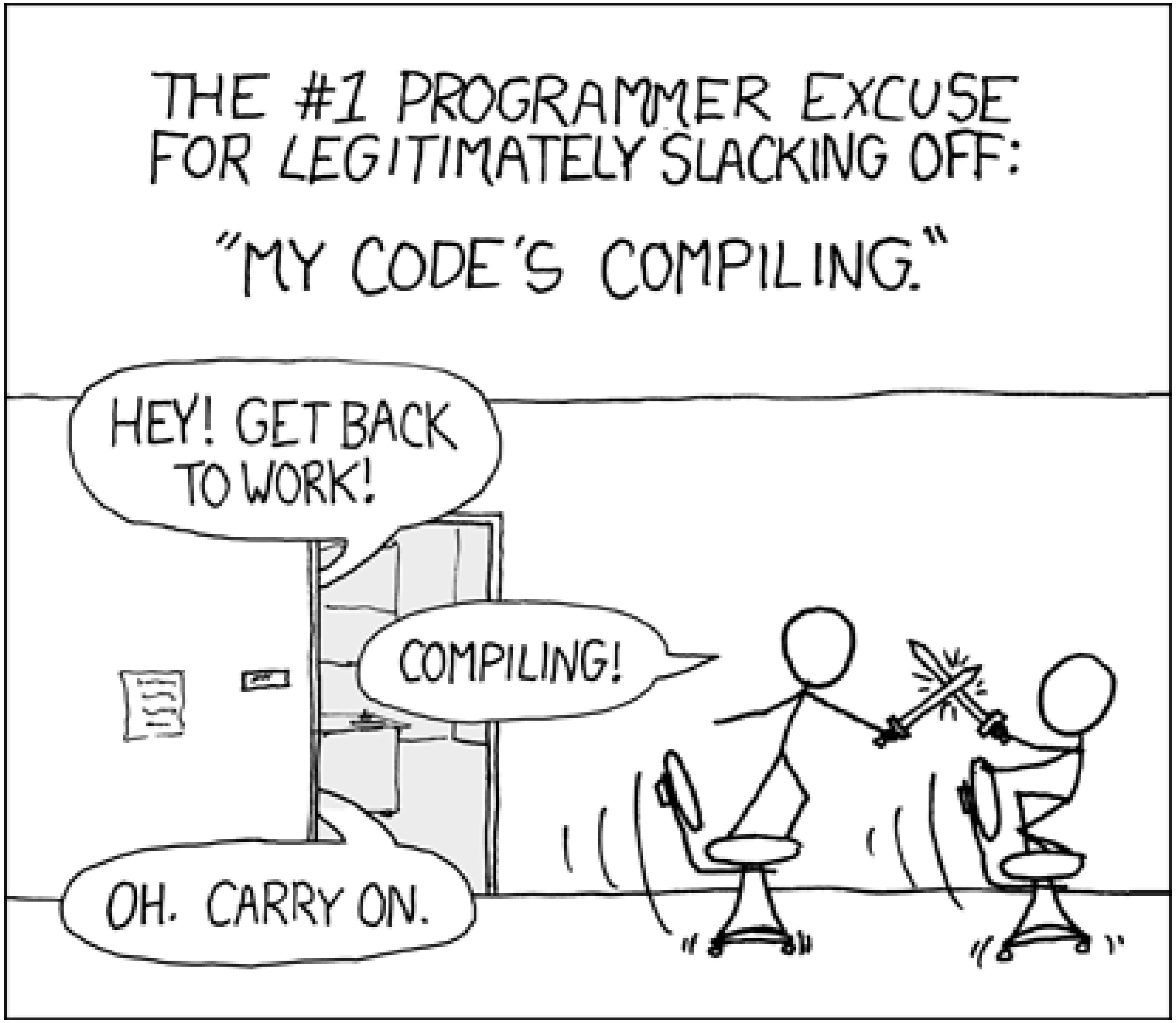
Prediction: Go will become the dominant language for systems work in IaaS, Orchestration, and PaaS in 24 months. #golang

RETWEETS 86

FAVORITES 56

7:00 AM - 11 Sep 2012

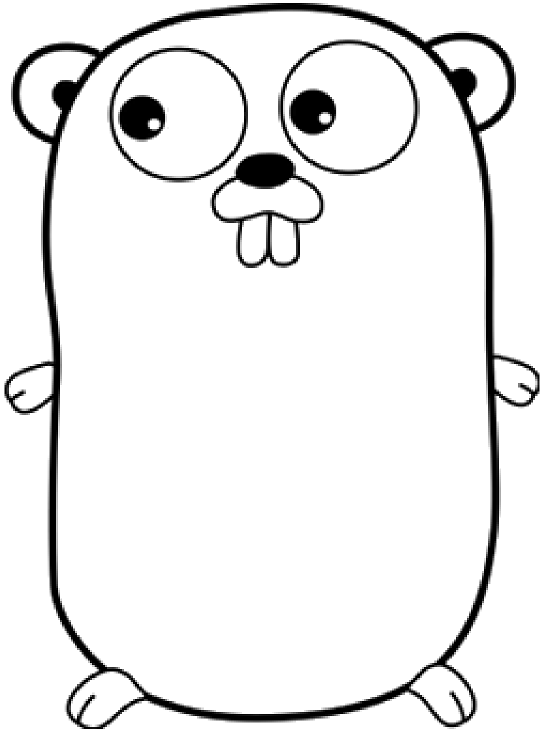
What is Go?



THE #1 PROGRAMMER EXCUSE FOR LEGITIMATELY SLACKING OFF:
"MY CODE'S COMPILING."
HEY! GET BACK TO WORK!
COMPILING!
OH. CARRY ON.

History

- Designed @ Google in 2007
- Open sourced in 2009
- Go 1 released in early 2012



Gopher by Renée French(http://golang.org/doc/gopher)

Background

Concurrent

Designed to run at scale

Created by:

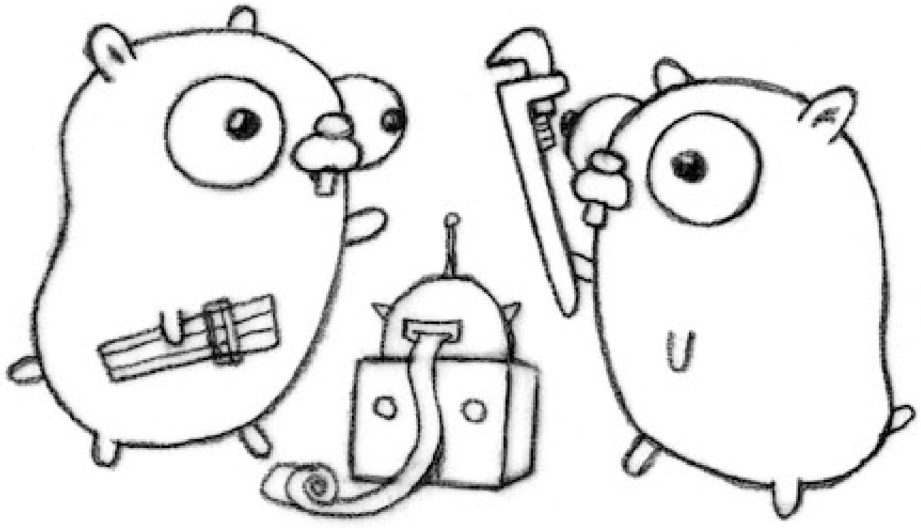
- Rob Pike and Ken Thompson (Unix, Plan 9, C, UTF-8)
- Robert Griesemer (Java HotSpot, V8, Chubby)

Why does Go exist?

Why does Go exist?

- Eliminate slowness
- Improve effectiveness
- Maintain (and improve) Google's scale

Designed by and for people who write large software systems



Gopher by Renée French (http://golang.org/doc/gopher)

Wait a minute… I am a Java programmer, what's in it for me?

Commonalities

- C family (imperative, braces)
- Statically typed
- Garbage collected
- Memory safe (nil references, runtime bounds checks)
- Variables are always initialized (zero/nil/false)
- Methods
- Interfaces
- Type assertions (instanceof)
- Reflection

Source: Go for Java Programmers by Sameer Ajmani (https://talks.golang.org/2015/go-for-java-programmers.slide)

Differences

- Programs compile to machine code. There's no VM.
- Statically linked binaries
- Control over memory layout
- Function values and lexical closures
- Built-in strings (UTF-8)
- Built-in generic maps and arrays/slices
- Built-in concurrency

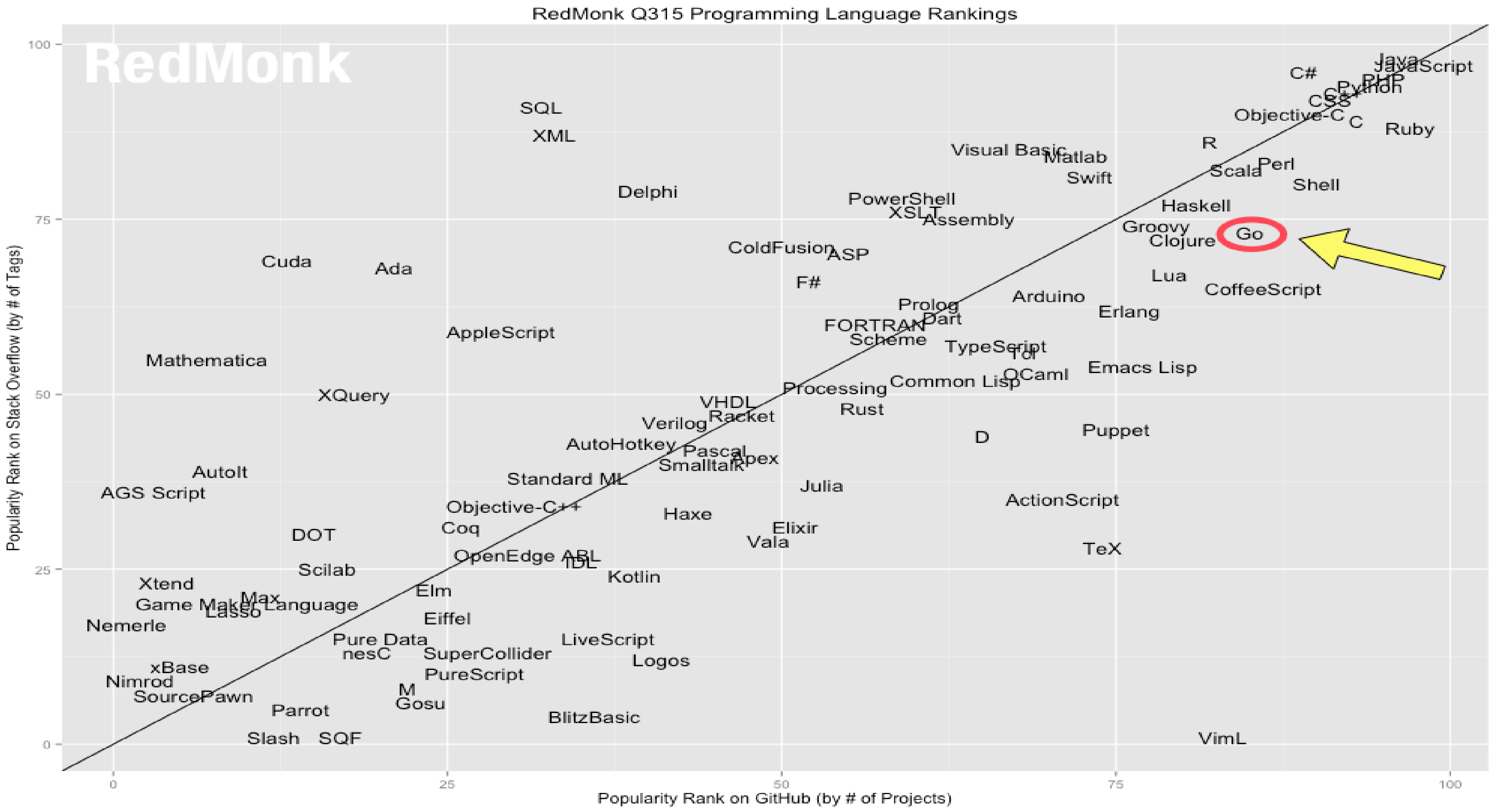
Source: Go for Java Programmers by Sameer Ajmani (https://talks.golang.org/2015/go-for-java-programmers.slide)

Surprises

- No classes
- No constructors
- No inheritance
- No final
- No exceptions
- No annotations
- No user-defined generics
- No official package manager
- No official IDE support

Source: Go for Java Programmers by Sameer Ajmani (https://talks.golang.org/2015/go-for-java-programmers.slide)

How popular is Go?



Source: The RedMonk Programming Language Rankings: June 2015 (http://redmonk.com/sogrady/2015/07/01/language-rankings-6-15)

Who uses it?

CLOUD FOUNDRY		Core OS	SendGrid
VAGRANT	DigitalOcean		
	Square		*And many more… (https://github.com/golang/go/wiki/GoUsers)*

Why is it being used?

- Simple -- only 25 keywords to remember
- Statically typed
- Cross-compile for many OSes and architectures
- Built-in, easy to use concurrency primitives
- Batteries included -- comprehensive standard library
- Superb tooling

Why is it being used?

- Open source (BSD 3-Clause license)
- Designed for large teams (e.g. open source projects)
- Easy deployment
- No dependency chaos... a single executable binary
- Fast builds and execution

So how does this relate to Predix?

- Cloud Foundry History
- TCP Router
- Components in Predix
- Buildpacks & Support

Why Go is a good choice for building apps in Predix?

- Smaller footprint
- Easier deployment
- Faster iteration cycle
- Cheaper to run
- Great JSON support

Show me some code!

Simple REST API

```go
func main() {
    http.HandleFunc("/api/v1/messages", func(w http.ResponseWriter, req *http.Request) {
        switch req.Method {
        case "POST":
            log.Println("got message:")
            defer req.Body.Close()
            io.Copy(os.Stdout, req.Body)
            w.WriteHeader(http.StatusAccepted)
        default:
            w.WriteHeader(http.StatusMethodNotAllowed)
            fmt.Fprintf(w, "%s is not allowed.", req.Method)
        }
    })
    log.Fatalln("error starting server:", http.ListenAndServe("localhost:8087", nil))
}
```

Run

JSON support

```
type Attendee struct {
    Name             string
    Email            string    `json:"email"`
    FoodPreference   string    `json:"food_preference,omitempty"`
    RegistrationDate time.Time `json:"registration_date"`
    Age              uint8     `json:"-"`
    realAge          uint8     `json:"real_age"`
}

func main() {
    attendee := &Attendee{
        Name:             "John Doe",
        Email:            "john.doe@ge.com",
        RegistrationDate: time.Now().AddDate(0, -2, 3),
        Age:              35,
        realAge:          50,
    }
    _ = json.NewEncoder(os.Stdout).Encode(attendee)
}
```

Run

How can I use Go in Predix?

Demo

A sample agent-collector implementation leveraging Go's strengths.

Littlebird: sl.ge/*littlebird (http://sl.ge/*littlebird)

Bigbird: sl.ge/*bigbird (http://sl.ge/*bigbird)

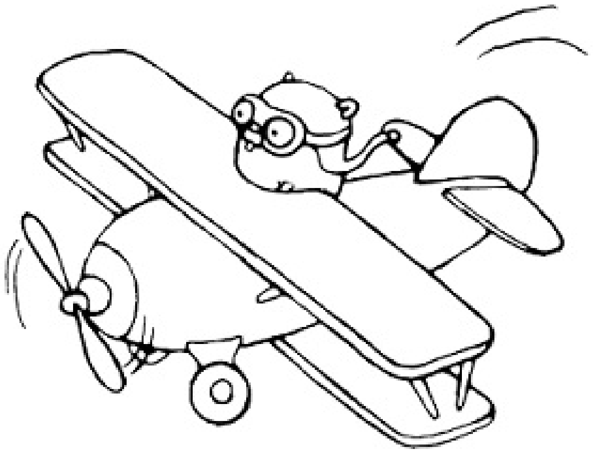
A few words on birdies...

- Can be used as a reference Predix application written in Go
- Idiomatic codebase with complete test suite
- Can be extended as a learning exercise
- *Entire commit history maintained since inception to facilitate learning*

Go ahead and add a receiver or a sink or two!
(Pull requests are more than welcome!)

So where is Go Go-ing?

- 1.5 is coming out soon
- Adding iOS Support
- Real-time GC Improvements
- Execution Tracing



Gopher by Renée French (http://golang.org/doc/gopher)

Where to Go from here?

- Go is fun
- Many companies are using it
- Great language for building Predix apps
- Cost effective
- Give it a Go!

Have questions? Want to know more? Reach out to the Go community in GE at The Go Programming Language (http://colab.ge.com/dashboard/canvas/v/525d9e2284429fa766000035) canvas.

Some Useful Links

Go homepage - http://golang.org (http://golang.org)

A Tour of Go - http://tour.golang.org (http://tour.golang.org)

The Go Playground - http://play.golang.org (http://play.golang.org/)

GoDoc - http://godoc.org (http://godoc.org/)

Go By Example - https://gobyexample.com (https://gobyexample.com)

State of Go - http://talks.golang.org/2015/state-of-go-may.slide (http://talks.golang.org/2015/state-of-go-may.slide)

Developer Zone…

A place for Developers to learn, network, relax, and have fun

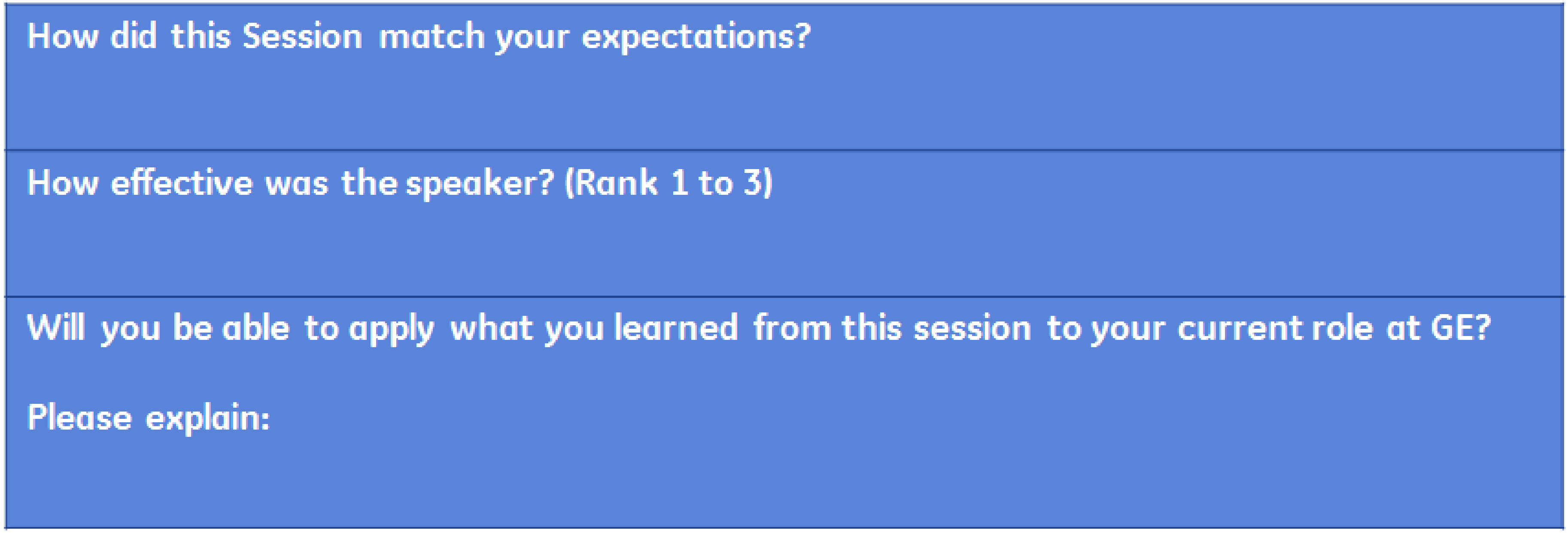
Hours: Tues. & Wed. 7pm-11pm

- **Predix SME's**
  - Hear from our Predix experts and ask questions
- **Performance Monitoring**
  - Hear about the tools and methodologies to benchmark apps deployed on Predix
- **Coding DOJO**
  - LIVE and interactive paired programming coding demonstration
- **Engage**
  - Interact with peers, try your hand at ping pong, shoot some pool, have fun!

STC Breakout Session Survey

Don’t forget to complete the survey after this breakout session!

Go to your Mobile App, click on the session you just attended, and complete just 3 survey questions:



How did this Session match your expectations?

How effective was the speaker? (Rank 1 to 3)

Will you be able to apply what you learned from this session to your current role at GE?

Please explain:

Disclaimer

General Electric reserves the right to make changes in specifications and features, or discontinue the product or service described at any time, without notice or obligation. These materials do not constitute a representation, warranty or documentation regarding the product or service featured. Illustrations are provided for informational purposes, and your configuration may differ.
This information does not constitute legal, financial, coding, or regulatory advice in connection with your use of the product or service. Please consult your professional advisors for any such advice.

GE, Predix and the GE Monogram are trademarks of General Electric Company.

Thank you

Cem Ezberci
Principal Architect, GE Software
cem@ge.com (mailto:cem@ge.com)

Fred Schults
Principal Architect, GE Software
Fred.Schults@ge.com (mailto:Fred.Schults@ge.com)