The Perils of Unit Testing Promises

Royi Hagigi

@rhagigi

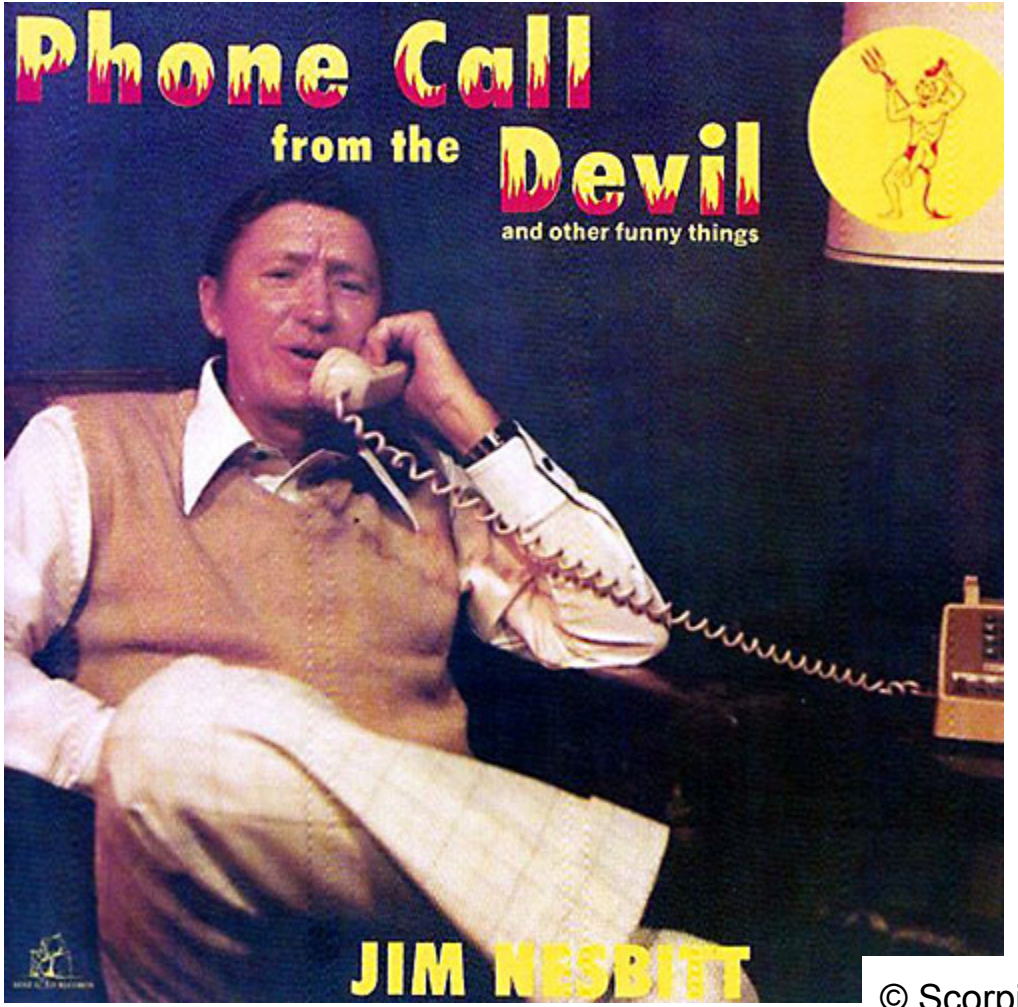

ONE LANE
ROAD
AHEAD

What to do?

Photo credit: bitzcelt / Foter / CC BY-NC-ND

```
function main(){
  getSomethingAsynchronously(someParam, someOtherParam, function(result) {
    nowLetsDoSomethingWith(result);
  });
}
```

problem?



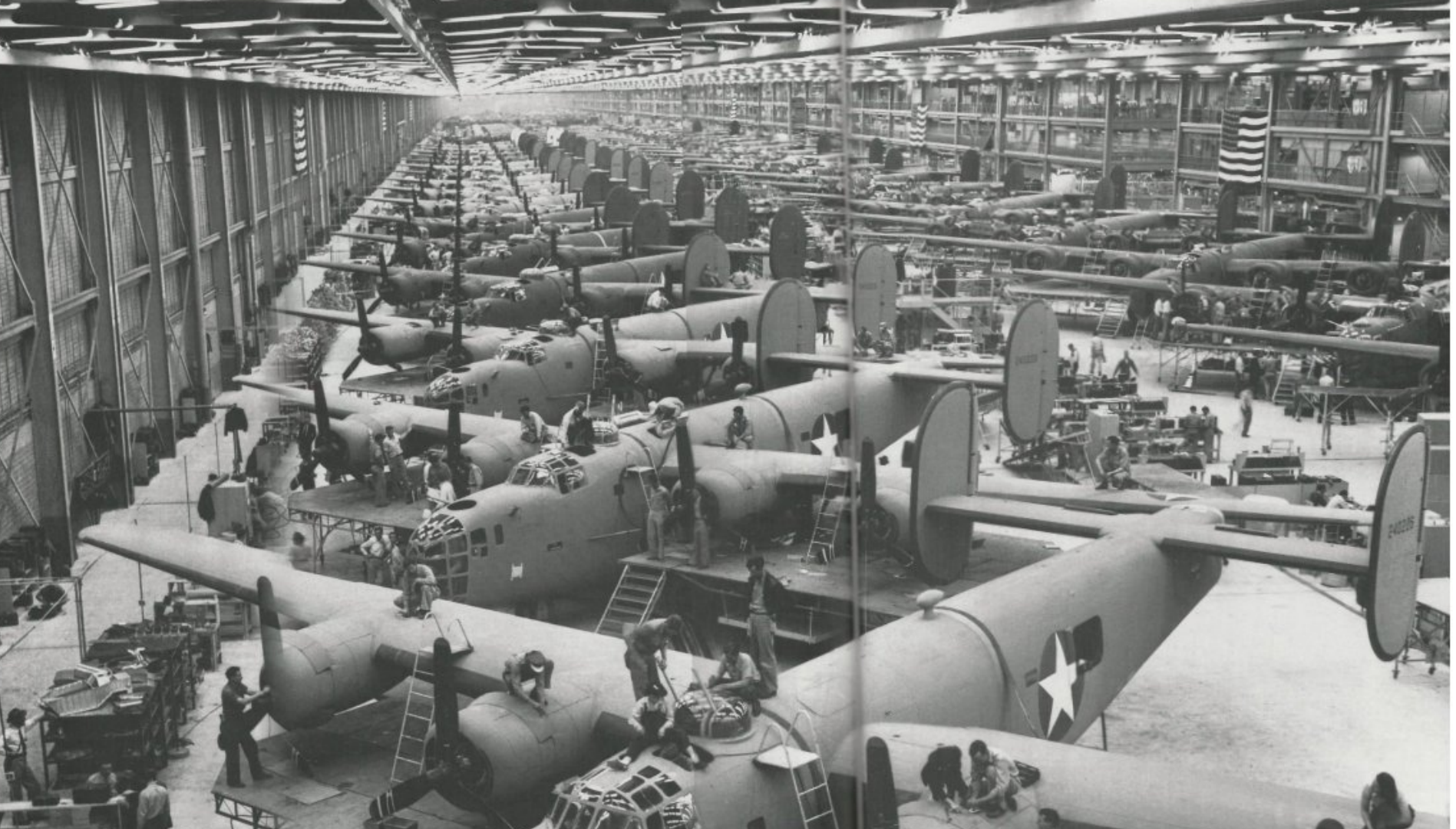
But Royi...

- Name your callbacks
- Avoid nesting inline callbacks (if you can help it)

What else?

[Code sample]

This is not ideal!



[Code sample]

Eww...
can’t we make this cleaner?

Promises 101

What is a promise?

- Lets you treat callbacks as objects
  - OOP approach to callbacks
- Returns a promise that will eventually be resolved… or rejected
- Chainable
- Reusable

example

```
var repoPromise = githubApi.getRepo(repoName);

repoPromise.then(function (repo){
    console.log(“watchers: ” + repo.watchers_count);
});
```

example

```
var repoPromise = githubApi.getRepo(repoName);

var addWatcherPromise =
    repoPromise.then(function (repo){
        console.log("watchers: " + repo.watchers_count);
    }).then(function (repo){
        return githubApi.addWatcher(repo, 'watcherName');
    });
```

example

```
var repoPromise = githubApi.getRepo(repoName);
repoPromise.then(function (repo){
    return githubApi.addWatcher(repo, ‘watcherName’);
}).done(function(){
    console.log(‘Successfully added a watcher’);
}, function(error) {
    console.error(error);
});
```

From node callbacks to Promises

- Q offers convenience functions
  - Q.denodeify
    - Turns node.js function (error, callback) into function that returns a promise
  - Q.ninvoke/nfcall
    - Calls node.js function/method and returns a promise

[Code sample]


REUSE
REDUCE
RECYCLE

BUT… what’s actually going on?

aka… deferreds 101

Some best practices

- If you’re gonna use promises, use promises
  - One size fits all -- only the deepest parts of your application need to wrap callbacks, everything else uses/composes/returns promises.
- Treat the “promise” like the value it promises, you can pass it around to places that need it
- .then()’s should always return a value.
- **End chains with a .done()!**

Aren’t I missing something...

As any good code reviewer would say...

Where are the tests!?!?!?

For the purposes of this demo...

What’s the difference?


12
1
2
3
4
5
6
7
8
9
10
11
Elgin

The results of your test aren’t done in the current Tick of the event loop.

[Code sample]

How do we test a function that calls promise functions?

How do we test a function that wraps a callback?

What about functions that will asynchronously respond to requests? (Server Side Controllers)

Another approach to mocking/DI...

Useful Libraries

https://github.com/domenic/mocha-as-promised

With “Chai as promised”, adds useful “thenable” asserts.

ex. expect(promise).to.**eventually**.equal (‘someValue’)

The end, I promise.

Questions, Comments Concerns?