

How to Bring Down Giants

Planning and Executing Epics, the Agile Way



HELLO!



Stjepan Rajko
stjepanr@axosoft.com
[@dancinghacker](https://twitter.com/dancinghacker)

Refactoring

Database
Transactions

SettingsUI

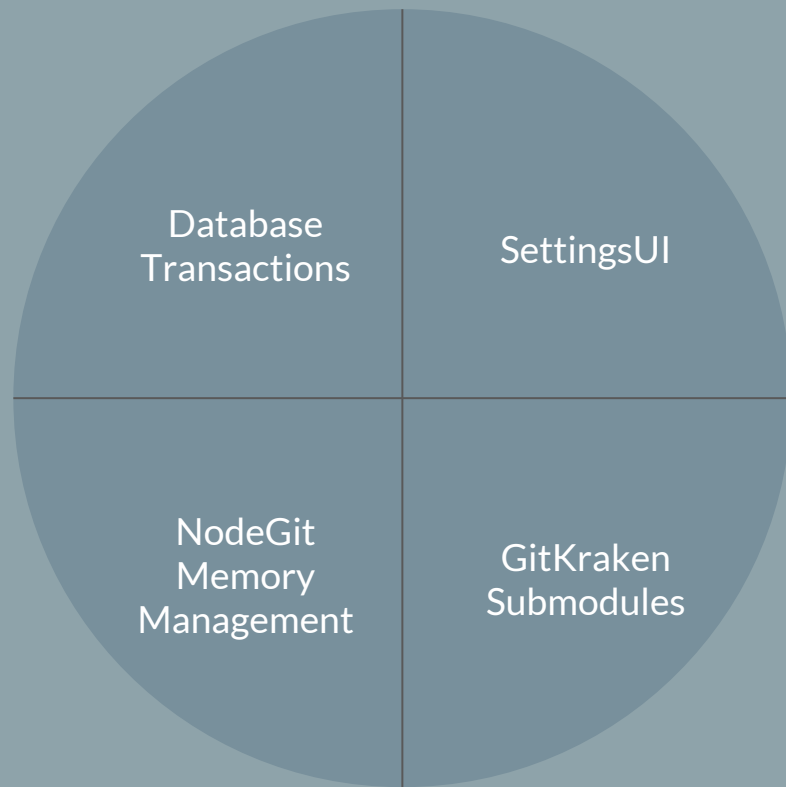
Features

NodeGit
Memory
Management

GitKraken
Submodules

Right

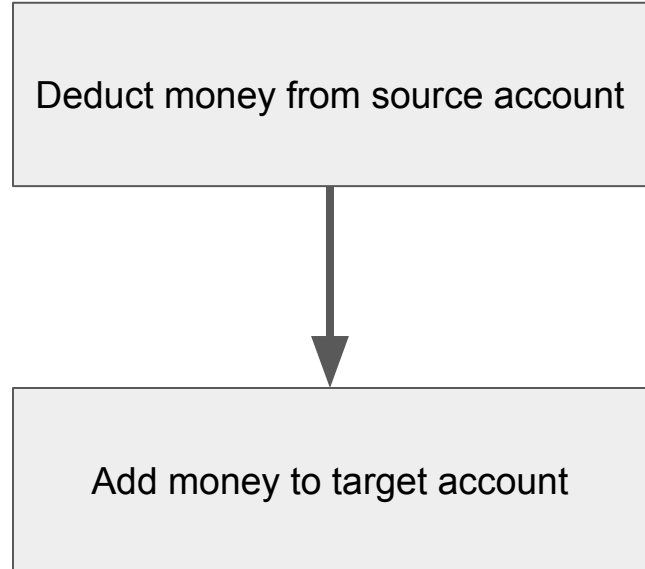
Wrong



Database Transactions

Refactoring gone right

DB Transactions 101:
Unit of work, composed of
multiple operations



Problem #1

```
DeductMoney(int accountId, int amount, TransactionHelper transaction=null)
{
    // if we forget to pass transaction, GetAccount may block
    var account = GetAccount(accountId, transaction);

    account.Deduct(amount);

    // if we forget to pass transaction, Save may block
    account.Save(transaction);
}
```

Problem #2

```
TransferMoney(..., TransactionHelper transaction=null)
{
    var makeNewTransaction = transaction == null;
    if (makeNewTransaction) {
        SqlHelper.BeginTransaction(connectionString, out transaction);
    }
    try {
        ...
    }
    catch {
        SqlHelper.RollbackTransaction(transaction);
        throw;
    }
    if (makeNewTransaction && SqlHelper.CommitTransaction(transaction)) {
        throw new SqlTransactionException();
    }
}
```

Problem #3

```
Save(TransactionHelper transaction == null)
{
    if(transaction != null) {
        SQLHelper.Execute(transaction, "UPDATE ACCOUNTS... ");
    } else {
        SQLHelper.Execute(connectionString, "UPDATE ACCOUNTS... ");
    }
}
```

Step #1:

```
Execute(Transaction transaction, string sql)
{
    if(transaction != null) {
        SQLHelper.Execute(transaction, sql);
    } else {
        SQLHelper.Execute(connectionString, sql);
    }
}
```

Simplify common usage

Step #1:

```
Save(TransactionHelper transaction=null)  
{  
    Execute(transaction, "UPDATE ACCOUNTS...");  
}
```

68 changed files with **569 additions** and **1,037 deletions**.

Step #2:

```
TransferMoney(..., TransactionHelper transaction=null)
{
    var makeNewTransaction = transaction == null;
    if (makeNewTransaction) {
        SqlHelper.BeginTransaction(connectionString, out transaction);
    }
    try {
        ...
    }
    catch {
        SqlHelper.RollbackTransaction(transaction);
        throw;
    }
    if (makeNewTransaction && SqlHelper.CommitTransaction(transaction)) {
        throw new SqlTransactionException();
    }
}
```

Simplify more common usage

Step #2:

```
void TransferMoney(int accountId, int amount, TransactionHelper transaction=null)
{
    using (Transaction(transaction)) {
        ...
    }
}
```

35 changed files with **1,015 additions** and **1,916 deletions**.

Step #3:

```
void TransferMoney(int accountId, int amount, TransactionHelper transaction=null)
{
    // now stores active transaction per thread+connectionString:
    using (Transaction(transaction)) {
        ...
    }
}

Save(TransactionHelper transaction == null)
{
    // now checks whether transaction matches the stored active transaction
    Execute(transaction, "UPDATE ACCOUNTS... ");
}
```

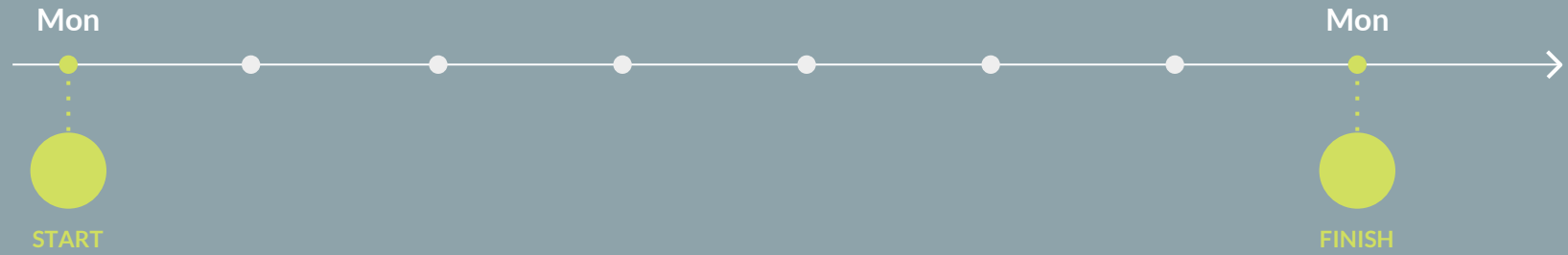
Step #4:

```
TransferMoney(int accountId, int amount)
{
    using (Transaction()) {
        ...
    }
}

Save()
{
    Execute("UPDATE ACCOUNTS... ");
}
```

70 changed files with **523 additions** and **528 deletions**

The timeline



Lessons learned:

- Find a step in the right direction that stands on its own
- START
- REPEAT
- At any point, you can STOP if needed
- Also: LISTEN TO YOUR USERS

SettingsUI

Refactoring gone wrong

System Settings

General

- System Labels
- Field Names
- Details Panel
- Axosoft API Settings
- SMTP Server
- Localization

Main Tab Visibility

Select the tabs that should be visible on the main window

<input checked="" type="checkbox"/> Defect Backlog	<input checked="" type="checkbox"/> Work Items	<input checked="" type="checkbox"/> T
<input checked="" type="checkbox"/> Help Desk	<input checked="" type="checkbox"/> Wiki	<input checked="" type="checkbox"/> E
<input checked="" type="checkbox"/> Release Planner		

Card View Progress Behavior

When clicking on the item's progress bar:

Do nothing
 Quick edit durations
 Add work log

Commenting System

Maximum comment length (0 for unlimited): Ask for c

2000 Require e

Detail View Settings

Settings: Customer Portal

General Settings

- Defect Backlog Grid Settings
- Work Items Grid Settings
- Tasks Grid Settings
- Help Desk Grid Settings
- Search Fields
- Appearance Settings
- Security & Customer Defaults
- Notification Settings
- External Authentication
- External Authentication Cookie
- Other Cookies
- Embedded Portal
- Embedded Headers

General Settings

Title for Customer Portal:

Axosoft Portal

Allow customers to self register
 Require customers to be approved before access is granted
 Send email notification when a new user registers

Notify the following email(s):

portal@axosoft.com

Allow customers to vote on items
 Allow customers to view and add comments to wiki pages

Notification Settings

Give customers adding items the o

Good people, good intentions...

1. Implement Customer Portal Settings from scratch
2. Implement System Settings - copy/pasted from System Settings
3. Roll our own component to support the two similar settings pages



Bad idea #1

The SettingsUI component

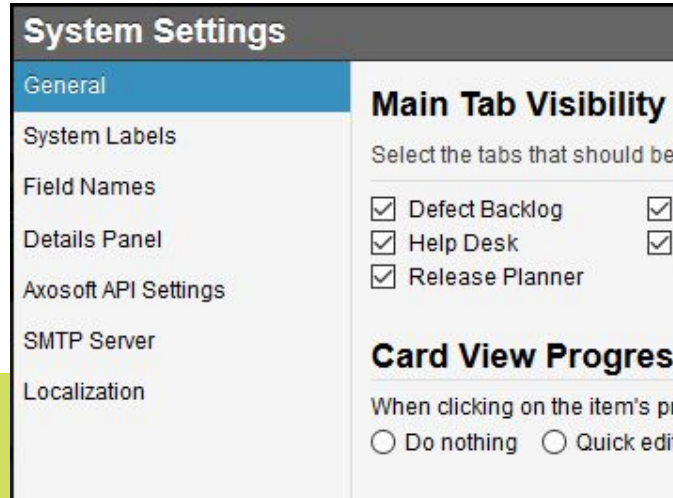
- Gets data from the server

```
{
  types: {
    defects: {
      enabled: true, ...
    }, ...
  }, ...
}
```

- Binds html to data

```
<span data-display-requires="types.defects">
  <input type="checkbox" data-id="types.defects.enabled"/>
  <label>Defect Backlog</label>
</span>
```

- Sends updated data to the server



Problem

As SettingsUI expands to support more and more pages,
it becomes an unmaintainable mess

Good people, good intentions...

	SettingsUI	FormHelper
Copy →	<ul style="list-style-type: none">● Gets data from the server	<ul style="list-style-type: none">● Gets data from the server
/		
Pasted →	<ul style="list-style-type: none">● Binds html to data	<ul style="list-style-type: none">● Binds html to data using Knockout.js
	<ul style="list-style-type: none">● Sends updated data to the server	<ul style="list-style-type: none">● Sends updated data to the server

Bad idea #2

Good people, good intentions...

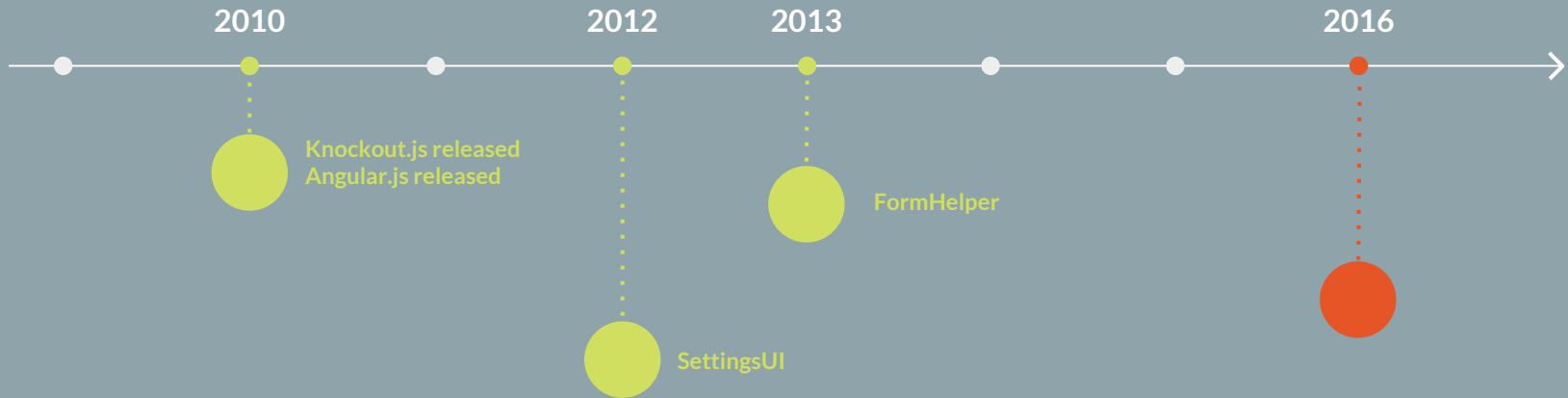
No migration plan

Bad idea #3

Problem now

- One bad component
- One better component
- Two components to deal with

The timeline



Lessons learned:

- Use external components
- OR: Open-source your component
- Have a full migration plan

GitKraken

Submodules

Feature gone wrong

The screenshot shows the GitKraken interface with a commit history on the right and a diff view on the left. The commit history includes entries like "Fix un/stageall and stashing", "Keep rename detection stage/unstage all", "Bump to version 0.1.40", "Merge pull request #597 from johnhaley81/fix-dispat...", "Merge pull request #594 from Mr-Wallet/nicer-ref-nam...", "Fix 'waitFor' bug in dispatcher", "Merge pull request #596 from johndavidsparrow/gh-p...", "Revised custom variable script and switch", "Merge pull request #595 from johndavidsparrow/gh-p...", "Resolved edge case where RefNodes could overlap", "A universe removal of in-app invite wording", "Bump to version 0.1.39", "Merge pull request #591 from Mr-Wallet/fix-graph-ref...", "Merge pull request #590 from srajko/dlv-be-gone", "Merge pull request #588 from Mr-Wallet/friendlier-app...", "Merge pull request #588 from Mr-Wallet/nicer-ref-nam...", "Merge pull request #589 from johndavidsparrow/gh-p...", "Merge pull request #592 from implausibler/NSFW", "JS tidy up in form-validation.js", "Javascript update for Universe", "A universe page", "added maxwait to updateworkdir debounce", "Update NSFW for memory leak", "Fix NSFW segfault", "Fix flickering GraphItemColumn every time ... 6 days ago", "Preventing page reload on default pull click", "Eliminate console spam when conflicts exist in a statel...", "Upgrading to react-bootstrap v0.24.5".

The diff view on the left shows changes to package.json, including updates to the version field from "0.1.39" to "0.1.40" and "5.1.11" to "5.1.12".

Main Repository

Submodule

Submodule



Can this be done in
two weeks?

Hamid Shojaee

V.P. Product

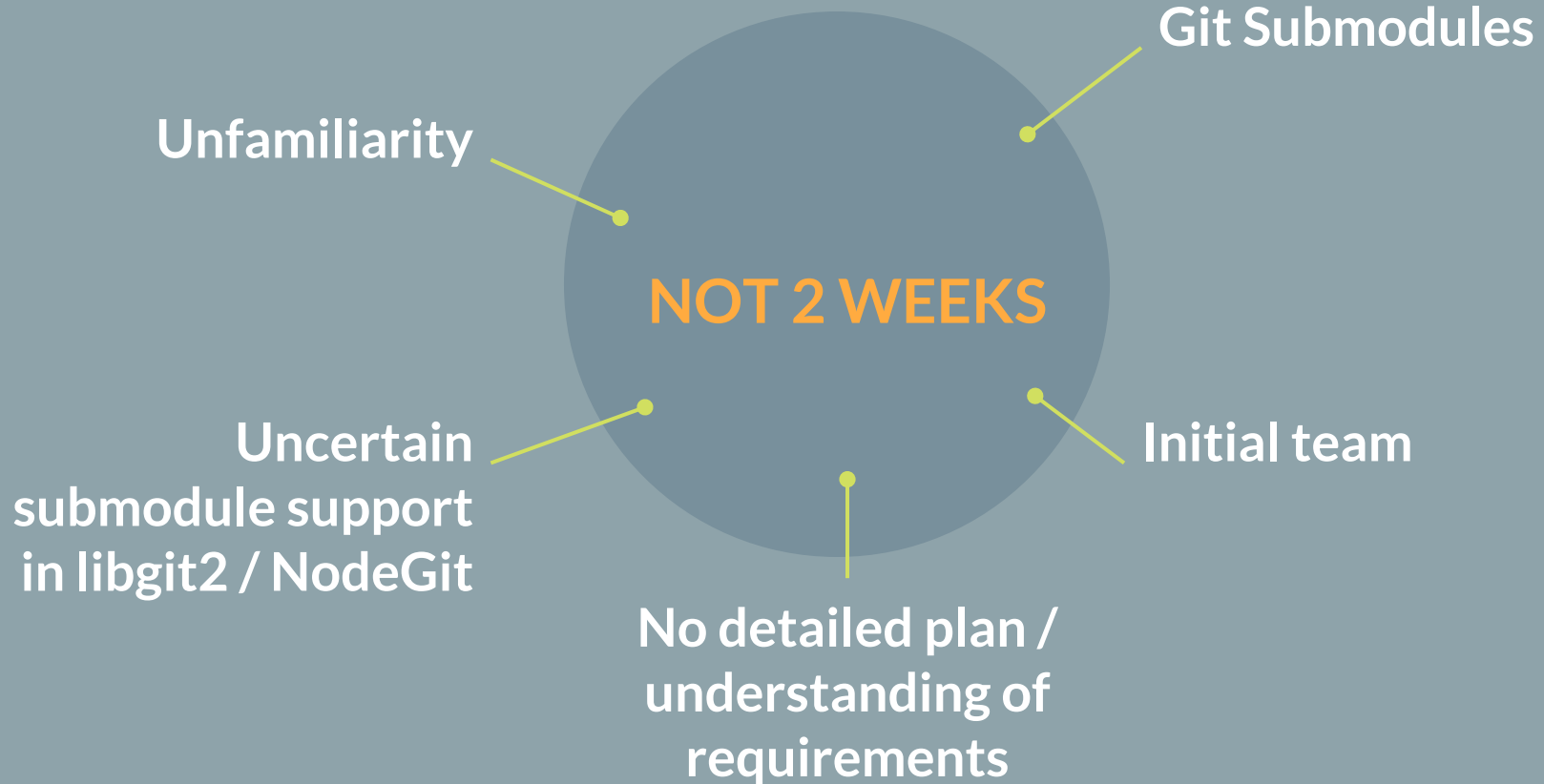


Sure!

Me

Idiot

It took 2+ months



Lessons learned:

- THINK before you estimate
- When wrong, STOP
- Rethink / replan
- Resume
- Or: DON'T resume

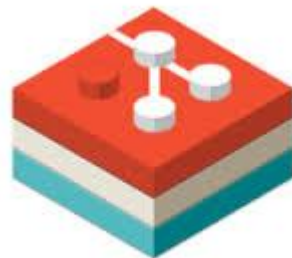
NodeGit Memory Management

Feature going right



NodeGit (JavaScript)

libgit2 (C)



The challenge

```
git_repository *repository;  
git_commit *commit;  
git_signature *signature1, *signature2;  
git_oid oid;  
git_oid_fromstr(&oid, "123456789...");  
  
git_repository_open(&repository, "/path/to/repo");  
git_commit_lookup(&commit, repository, oid);  
signature1 = git_commit_author(commit);  
git_signature_default(&signature2, repository);  
  
git_commit_free(commit);  
git_signature_free(signature2);  
git_repository_free(repository);
```

```
repository = NodeGit.Repository.open("/path/to/repo");  
commit = repository.lookupCommit("123456789...");  
signature1 = commit.author();  
signature2 = repository.defaultSignature();
```

C

vs.

JavaScript

Problem

- If you free incorrectly, code starts to crash

Current Solution

Time	5165282
Oid	3299899
Reference	422227
Object	285935
Remote	145311
TreeEntry	12801
DiffDelta	8146
DiffFile	5759
Cred	3522
Cert	3522
StatusEntry	2823
IndexTime	2168
Revwalk	1408
Strarray	1234
Index	1167
StatusList	1102
Config	572
ReflogEntry	131
Diff	98

Don't free!

Plan of attack

- One type at a time
- Add memory management mechanisms as needed



Effort per type

Lessons learned:

- Effort per type is fickle
- Consistent progress is good

Main takeaways:

- Find a good first step to tackling the giant
- Develop plan to defeat giant completely
- Don't create giant problems
- When realizing you are facing a giant, STOP

THANK YOU!



Stjepan Rajko

stjepanr@axosoft.com

www.axosoft.com