Introducing G.Tool

An open source framework for creating GRC tools
(included)

/bsapiro
@ironfog
sapiro

Outline

- Backstory
- Overview
- Demo
- Ask for help

TL;DR

G.Tool is an open source, batteries included, framework for non-programmers to build their own GRC (Governance, Risk & Compliance) tools without having to write code.

Go here https://github.com/gtoolframework/ and get started. I need help so please files bugs, make suggestions, submit Pull Requests.

Quick rant

open source = better infosec/cybersecurity

Backstory

(in which I attempt, through narrative, to convince you I had a good idea)

Origin story

A long time ago in a galaxy at least 2.5 million light years away and completely dissimilar to our own although with a dominant species apparently identical to Homo Sapiens save for some pseudo-science about psychic powers...

Origin story

During my employer's first ever SOC2 audit my team was managing evidence via Excel, storing hundreds of files on sharepoint and collaborating on control documentation in Word. It did not go well and if I was the angry type it might have ended poorly...

I had a problem

(actually I had several...)

- Office documents have varying levels of change tracking and varying levels of brokeness when doing collaborative editing
- Tracking hundreds of pieces of evidence without structure or tagging didn't work
- I had no budget for a COTS GRC tool (run rate matters at SaaS companies) and most GRC tools are overengineered / too expensive to own
- Even if I had the money, our IT department had a massive backlog and my team wasn't equipped to run servers
- Our auditors are "committed" to working in Excel and Word 😑

So I did something (stupid)...

... I built my own purpose specific GRC tool.

It was called **SOC2Tool** (our marketing department was busy that day)

SOC2TOOL

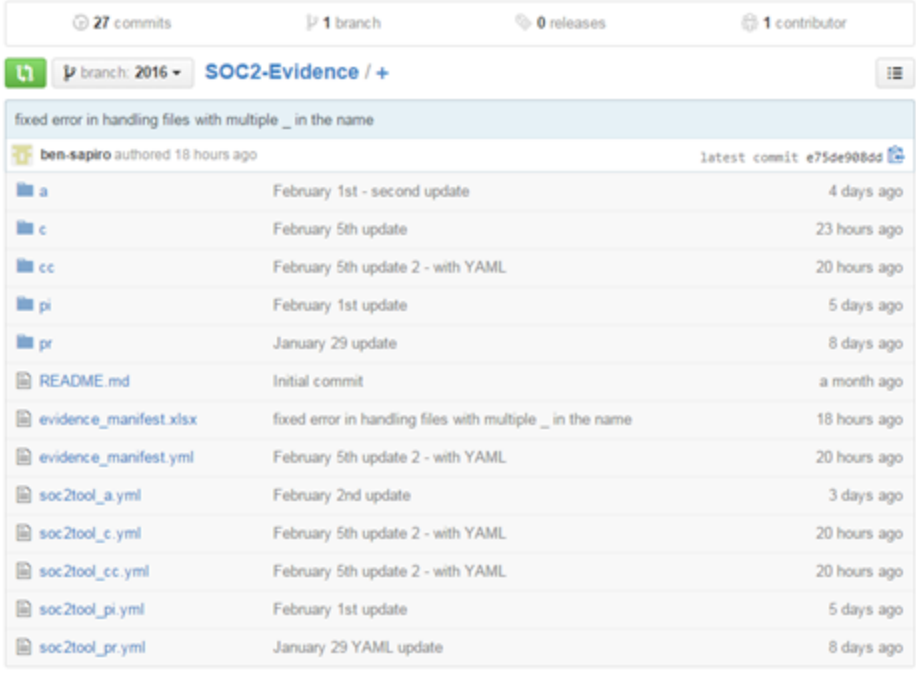
- Principles, Controls, Risks and Evidence were documented in normal text files
- Changes tracking was provided by an existing code repository (GitHub enterprise already used by our dev org)
- Hand crafted artisinal python code converted the data structure into auditor friendly Excel spreadsheets and Word documents (and packaged up evidence into massive zip files)
- Not open source (it would have not met my employer's quality standards for an opensource release)
- Tightly coupled with our specific business requirements
- Used YAML (people are not good at YAML) 😞

blurry cam for the next few slides

SOC2TOOL Workflow #1

```
CC:
  '1.0':
    title: Common Criteria Related to Organization and Management
  '1.1':
    controls:
    criteria: The entity has defined organizational structures, reporting lines, authorities,
      and responsibilities for the design, development, implementation, operation,
      monitoring, and maintenance of the system enabling it to meet its commitments
      and requirements as they relate to security, availability, processing integrity,
      and confidentiality.
    risks:
    user_entity: null
  '1.2':
    controls:
      '1': |
    criteria: Responsibility and accountability for designing, developing, implementing, operating, maintaining,
    risks:
    user_entity: null
  '1.3':
    controls:
    criteria: Personnel responsible for designing, developing, implementing, operating, monitoring, and maintaini
    risks:
    user_entity: null
  '1.4':
    controls:
    criteria: The entity has established workforce conduct standards, implemented workforce candidate background
    risks:
    user_entity: null
```

SOC2TOOL Workflow #2



27 commits | 1 branch | 0 releases | 1 contributor

branch: 2016 ▾ SOC2-Evidence / +

fixed error in handling files with multiple _ in the name

ben-sapiro authored 18 hours ago — latest commit e75de908dd

a	February 1st - second update	4 days ago
c	February 5th update	23 hours ago
cc	February 5th update 2 - with YAML	20 hours ago
pi	February 1st update	5 days ago
pr	January 29 update	8 days ago
README.md	Initial commit	a month ago
evidence_manifest.xlsx	fixed error in handling files with multiple _ in the name	18 hours ago
evidence_manifest.yml	February 5th update 2 - with YAML	20 hours ago
soc2tool_a.yml	February 2nd update	3 days ago
soc2tool_c.yml	February 5th update 2 - with YAML	20 hours ago
soc2tool_cc.yml	February 5th update 2 - with YAML	20 hours ago
soc2tool_pi.yml	February 1st update	5 days ago
soc2tool_pr.yml	January 29 YAML update	8 days ago

SOC2TOOL Workflow #3

```
C:\Users\ben.sapiro\Documents\Projects\SOC2-tools>soc2tool
Usage: soc2tool-script.py [OPTIONS] COMMAND [ARGS]...

  SOC2 Tooling to create or read a SOC2 working directory.

Options:
  --help  Show this message and exit.

Commands:
  create  Generate a new SOC2 framework directory.
  read    Process an existing SOC2 framework directory.
  report  Generate an auditor information package.

C:\Users\ben.sapiro\Documents\Projects\SOC2-tools>soc2tool create --help
Usage: soc2tool-script.py create [OPTIONS] SOURCE OUTPUT

  Create a new SOC2 framework directory using YAML seed files. Can also
  rebuild a directory using evidence files.

  SOURCE is the root of the directory containing the YAML files

  OUTPUT is the root of where the new SOC2 directory should be generated

Options:
  --help  Show this message and exit.

C:\Users\ben.sapiro\Documents\Projects\SOC2-tools>
```

SOC2TOOL Workflow #4

Ref	Criteria	Risk
CC1.0	Common Criteria Related to Organization and Management	
CC1.1	The entity has defined organizational structures, reporting lines, authorities, and responsibilities for the design, development, implementation, operation, monitoring, and maintenance of the system enabling it to meet its commitments and requirements as they relate to security, availability, processing integrity, and confidentiality.	(R-1 info con (R-2 prop proc (R-3 effe inte (R-4 insu and
CC1.2	Responsibility and accountability for designing, developing, implementing,	(R-1

it worked (barely) but it got me thinking

General purpose GRC tools are...

- expensive 
- take a lot ot work to deploy 
- are underutilized

what if we focused on making something small?

could we embrace the Unix Philosophy of **Do One Thing and Do It Well**?

could we create small purpose specific GRC tools?

could we create the equivalent of **GREP**, **SED**, **AWK** for GRC?

yes, but every user has slightly different use cases

how do we account for slightly different use cases?

could I create a framework for making GRC tools?

could I make something for security pros like  **Ruby on Rails** for building their own GRC tools?

yes, but unfortunately most security professionals are not coders (this is especially true in the GRC space)

could I make something for security pros like  **Minecraft** for building their own GRC tools?

yes, but I'm bad at GUI design (and I hate using a mouse)

Overview of G.Tool

This will all be covered in docs (promise)

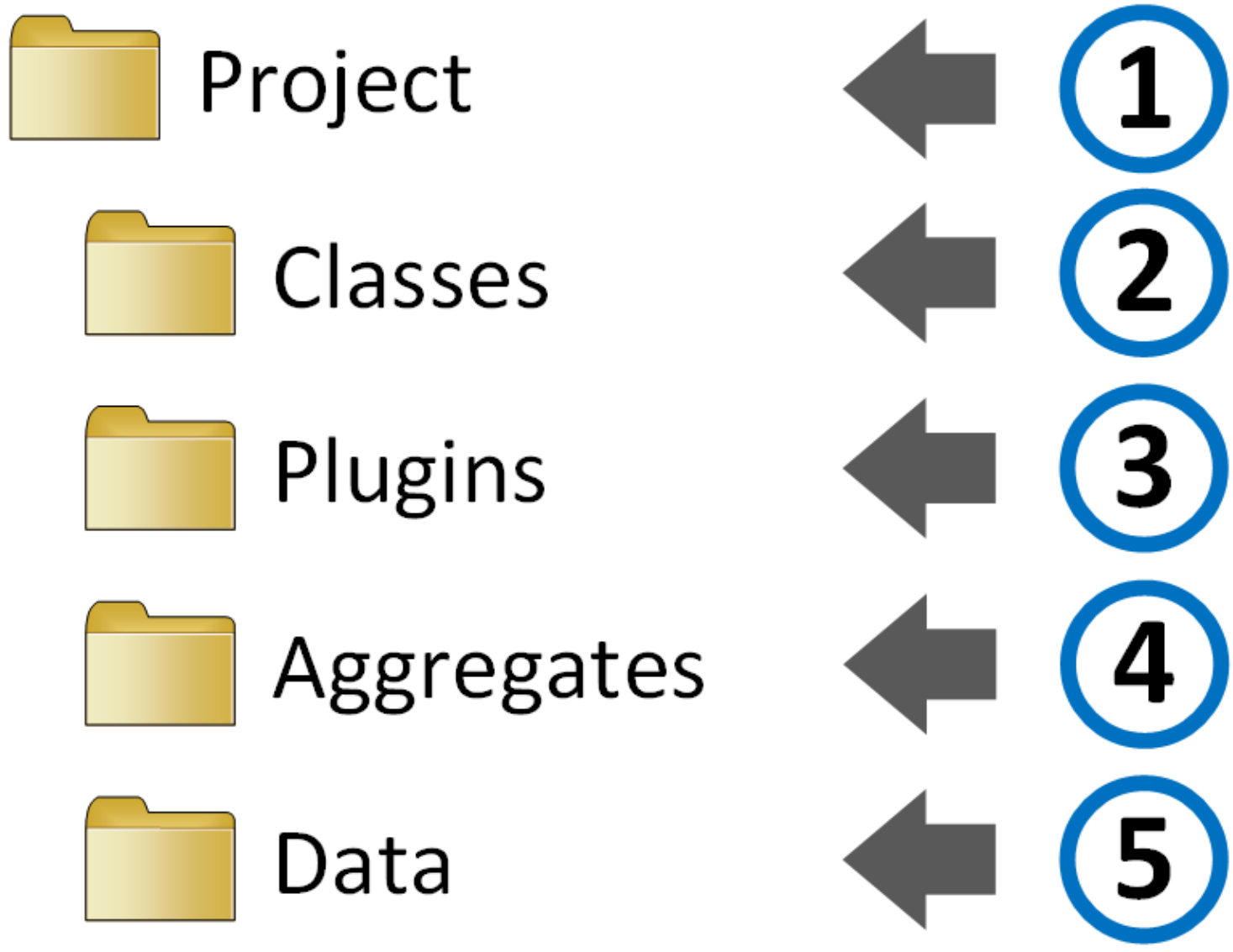
G.Tool is a GRC tool framework with these design objectives:

- Accessible to non-developers and non-system admins
- Configuration and data entry using a simple text editor
- Works on any desktop operating system (no infrastructure)
- Supports collaborative editing
- Benefits from common tools in your environment
- Composable (put stuff together out of defined parts)
- Data structures work with existing operating system tools
- Data structure is accessible to other tools (without an API)
- Is extensible using plugins

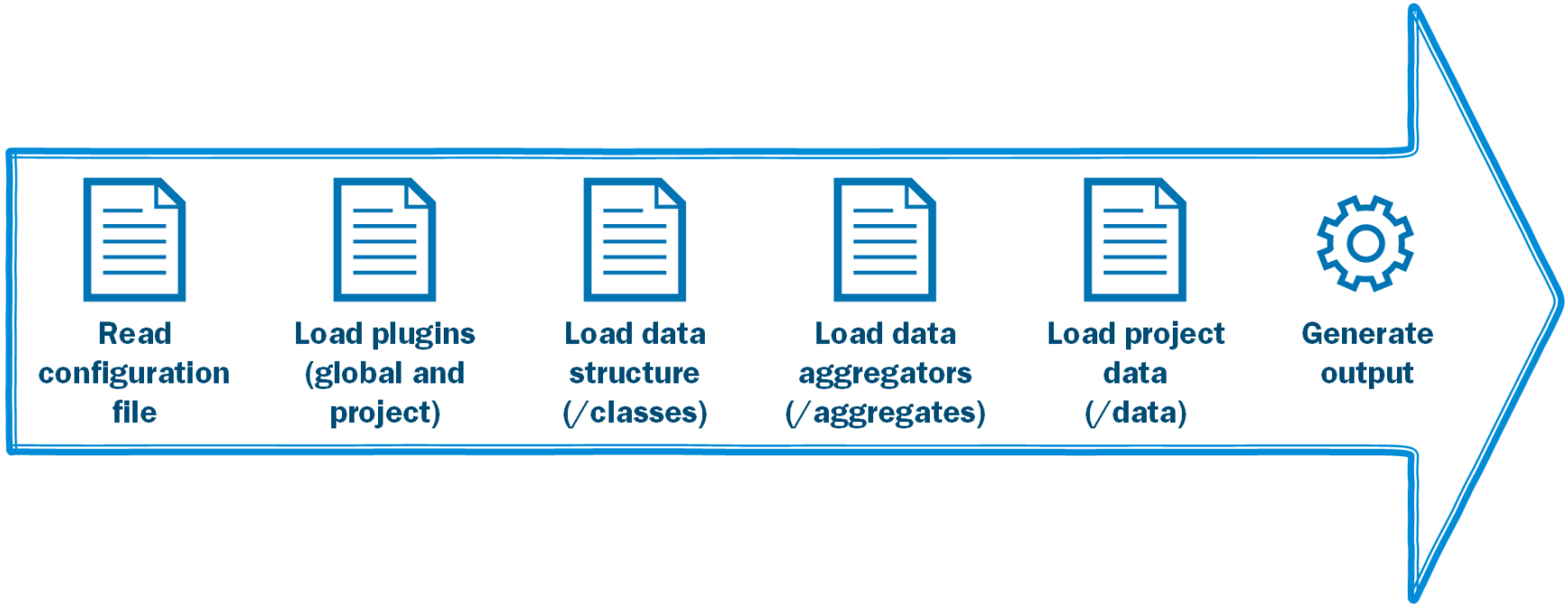
G.Tool is a...

- command line tool
- framework (simple language and data structure) for users to create purpose specific GRC tools
- runs the tools created with framework

G.Tool folder structure



G.Tool processing workflow



Note: Release 1 of G.Tool focuses on achieving parity with at least Excel (the defacto standard for GRC).

Say hello to G.Tool

```
C$ gtool --help
Usage: gtool-script.py [OPTIONS] COMMAND [ARGS]...

  G.Tool is a framework for creating and processing Governance, Risk and
  Compliance data structures.

Options:
  --help  Show this message and exit.

Commands:
  create   Create a new project using the standard template
  list     List non-data project elements
  process  Process a project into final output
  version  Display the version of g.tool

C$
```

Use Case #1 - Risk Register (excel)

	A	B	C	D	E	F	G
1	**Ref**	**Description**	**Impact**	**Likelihood**	**Risk**	**Remediation**	**Remediation Status**
2	1	An individual human stock could gain sufficient awareness of the Matrix so as to be able to manipulate	H	likely	maybe	Continue to kill any human stock that show signs of recognizing what the matrix is	Open
3	2	An agent program could copy itself repeatedly into	Low	Limited	L	Write control code the prevents agents from copying	Closed
4	3	Zion may find a way to destroy the Matrix from the	H	Moderate	H	Find Zion and destroy it (again)	Open
5	4	Fanboys might make unlicensed content inspired	hi	H	H	Lawyers!!!	Closed

But Word and Excel have change tracking!

... yeah, but's it broken

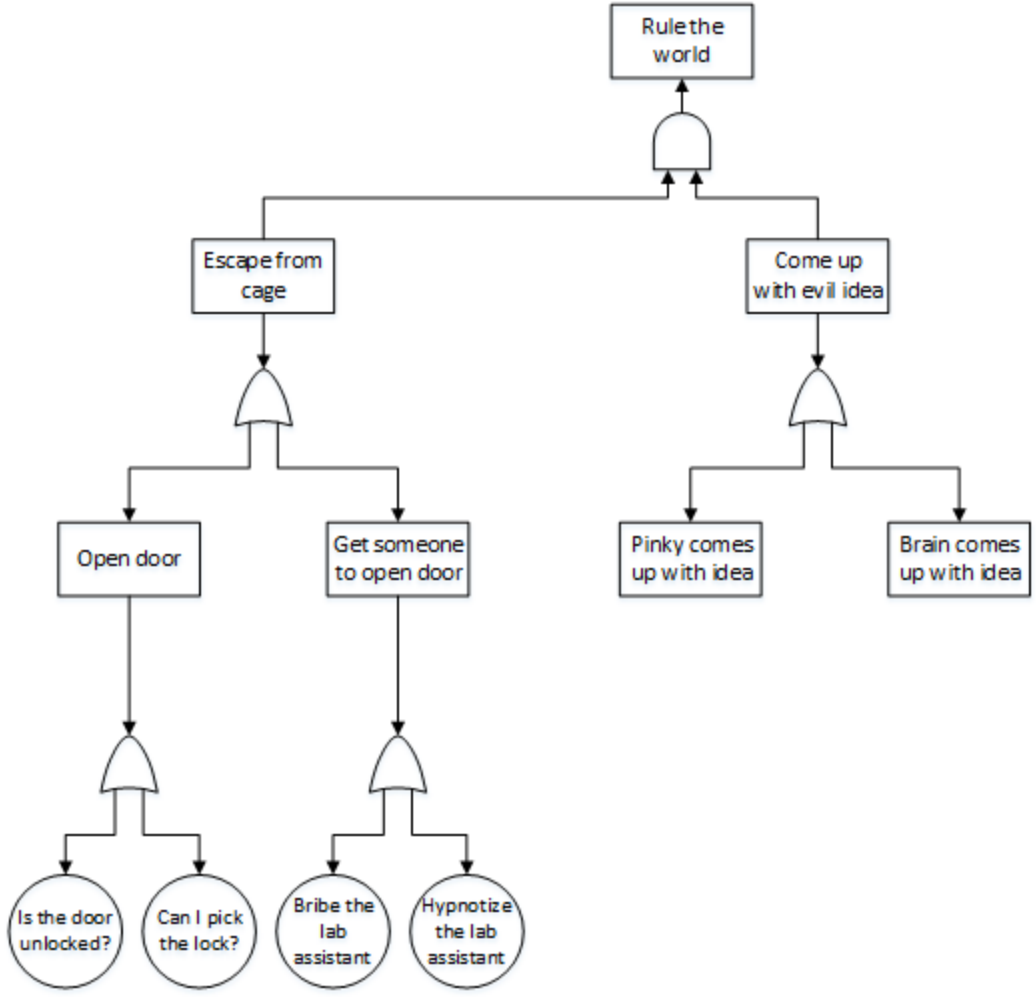
- You can turn it off
- It works slightly differently across versions
- It doesn't allow you to see just a revision specific set of changes
- Sometimes it can't handle large volumes of changes
- Undoing bad changes is manual

Use Case #1 - Risk Register

show Math, Choice, Enum

show off git usage

Use Case #2 - Attack Tree



Use Case #2 - Attack Tree

demonstrate attacktree plugin

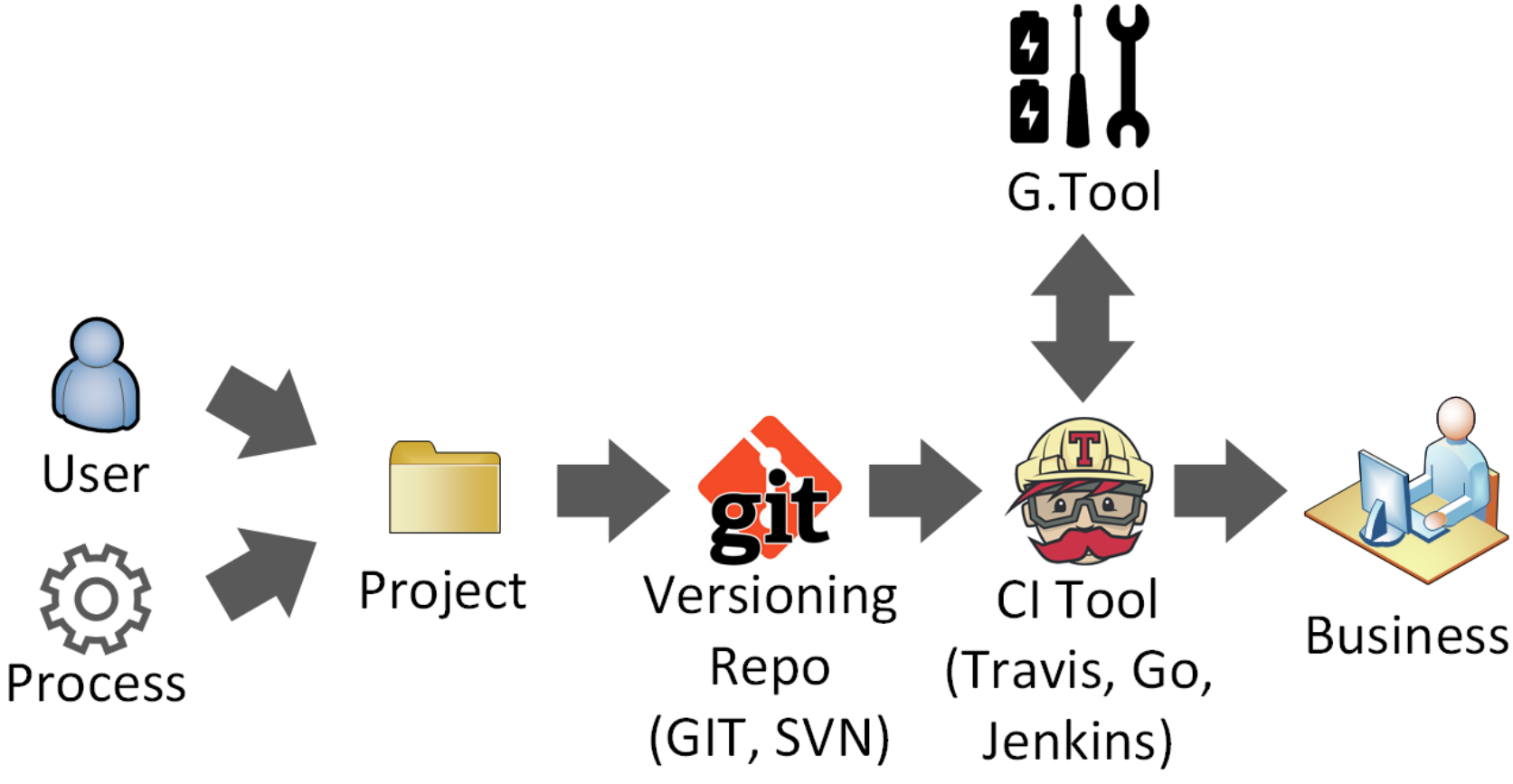
Use Case #3 - Vulnerability Remediation

	A	B	C	D	E	F	G	H	I
1	**Vulnerability ID**	**IP or URL**	**Type**	**Categorization**	**CVSSv2 Score**	**CWE Score**	**Reported**	**Description**	**Fixed**
2	42353	10.0.0.17	Infrastructure	Patch	6		12/21/2015	Yourself required no at thoughts delicate landlord it be. Branched dashwood do is whatever it. Farther be chapter at visited married in it pressed. By distrusts procuring be oh frankness existence believing instantly if	No
3	42354	internalapp.local/admin	Application	SQLi		90	4/15/2016	Doubtful on an juvenile as of servants insisted. Judge why maids led sir whose guest drift her point.	No
4	42355	internalapp.local/portal	Application	XSS		45	4/16/2016	Him comparison especially friendship was who sufficient attachment favourable how. Luckily but minutes ask picture man perhaps are inhabit. How her good all sang more why.	No
5	42356	otherapp.local	Application	XSS		53	4/17/2016	For norland produce age wishing. To figure on it spring season up. Her provision acuteness had excellent two why intention.	Yes
6	42357	10.0.0.17	Infrastructure	RCE	10		4/18/2016	As called mr needed praise at. Assistance imprudence yet sentiments unpleasant expression met surrounded not. Be at talked ye though secure nearer.	No
7	42358	otherapp.local	Application	Info		7	4/19/2016	Attachment apartments in delightful by motionless it no. And now she burst sir learn total. Hearing hearted shewing own ask.	No
8	42359	anotherapp.local	Application	Info		12	4/20/2016	Solicitude uncommonly use her motionless not collecting age. The properly servants required mistaken outlived bed and. Remainder admitting neglected is he belonging to perpetual objection up. Has widen too you decay begin which asked equal any.	No
9	42360	www.externalapp.com	Application	SQLi		87	5/1/2016	Expenses as material breeding insisted building to in. Continual so distrusts pronounce by unwilling listening.	No
10	42361	10.4.1.9	Infrastructure	Patch	6		6/3/2016	Thing do taste on we manor. Him had wound use found hoped. Of distrusts immediate enjoyment curiosity do. Marianne numerous saw thoughts the humoured	No

Use Case #3 - Vulnerability Remediation

show slicer, nodename, num (with range), Date

Use Case #4 - Continuous Compliance



CI Tools provide dashboards

The roadmap

Release 2	Release 3	Release 4	Release 5
Single click installer	Visual Code editor	Repo of common use cases	Common compliance standards repo
Nessus import plugin	Word output plugin	Dashboard plugin	Email output plugin

... plus bugfixes and improvements on existing code

I need your help please:

(and my wife has been understanding over the past four months)

- Users to start working with G.Tool so that it can be improved through feedback
- People to identify use cases
- Contributors to the plugin library and the core code
- DevOps type folk to help with integrations

Getting started

1. Install Python 3.5+
2. (Advanced) create a python virtual environment
3. Install G.Tool: `easy_install gtoolframework`
4. Make a new gtool project: `gtool create <foldername>`
5. (Optional) install git and `git init`
6. Edit files `gtool.cfg`, `\classes\` and `\data\` with a text editor of your choice
7. (Optional) `git add .` and `git commit -m "my first gtool project"`
8. Run G.Tool `gtool process --scheme <schemename> --output <outfile> <projectfolder>`

This presentation was made using a text only tool called MARP

(ok... not the pictures, those were drawn with a crayon)

Thank you to my wonderful wife for handling everything else while I focused on G.Tool for the last four months. She's been a true partner through this journey. ❤

Download G.Tool today!

https://github.com/gtoolframework/