

Presentations — Contents

1. [2019-Feb-13 SLG Presentation](#)
2. [2019 DrupalCon Drupal 8 Multisite](#)
3. [AI-Accelerated CMS Migration with Claude Code](#)
4. [DrupalCon 2022 Code+ Presentation](#)
5. [Introduction to Drupal](#)
6. [Drupal Multisite on a Dime](#)
7. [How to Manage Departmental Faculty and Staff Data](#)
8. [What I did at DrupalCon 2022](#)

From F to B

in Mozilla's Observatory Security Report

The **Trinity Technology Services** system administrators implement changes to Varnish to dramatically improve our Drupal website's security.

- Kyle Skrinak
- David Palmer
- Blaine Ott

<https://kyle.skrinak.com/presentations/2019-Feb-SLG.html>

The Problem

- ITSO sends TTS NetSparker report
- We break out the Drupal Security Kit module
- We create a test configuration and ask for another scan
- NetSparker is now busy, ITSO recommends Observatory

Mozilla Observatory

Observatory by Mozilla has helped over 125,000 websites by teaching developers, system administrators, and security professionals how configure their sites safely and securely.

Scan your site



- Don't include my site in the public results
- Force a rescan instead of returning cached results
- Don't scan with third-party scanners

- <https://developer.mozilla.org/en-US/observatory>

Original website results

Test Scores

Test	Pass	Score	Explanation	
Content Security Policy	✗	-25	Content Security Policy (CSP) header not implemented	i
Cookies	✓	0	All cookies use the <code>Secure</code> flag and all session cookies use the <code>HttpOnly</code> flag	i
Cross-origin Resource Sharing	✓	0	Content is not visible via cross-origin resource sharing (CORS) files or headers	i
HTTP Public Key Pinning	–	0	HTTP Public Key Pinning (HPKP) header not implemented (optional)	i
HTTP Strict Transport Security	✗	-20	HTTP Strict Transport Security (HSTS) header not implemented	i
Redirection	✓	0	Initial redirection is to HTTPS on same host, final destination is HTTPS	i
Referrer Policy	–	0	Referrer-Policy header not implemented (optional)	i
Subresource Integrity	✗	-50	Subresource Integrity (SRI) not implemented, and external scripts are loaded over HTTP or use protocol-relative URLs via <code>src="//..."</code>	i
X-Content-Type-Options	✗	-5	X-Content-Type-Options header not implemented	i
X-Frame-Options	✓	0	X-Frame-Options (XFO) header set to <code>SAMEORIGIN</code> or <code>DENY</code>	i
X-XSS-Protection	✗	-10	X-XSS-Protection header not implemented	i

Our First Attempt

```
### TTS Apache Hardening  
set resp.http.X-XSS-Protection = "1; mode=block";  
set resp.http.Strict-Transport-Security= "max-age=31536000; includeSubDomains";  
set resp.http.X-Content-Type-Options = "nosniff";
```

Oops



We inadvertently blocked fourth-level domains

Web Security x +

https://infosec.mozilla.org/guidelines/web_security#http-strict-transport-security

Apps Yahoo Weather Nagios - Linux Nagios - Errors Naigios - Warnings Nagios - Home Duke Mail vSphere Client MariaDB Docume... Nagios-Biology TTS Imported

moz://a home security fundamentals security guidelines improve this document discover firefox

HTTP Strict Transport Security

HTTP Strict Transport Security (HSTS) is an HTTP header that notifies user agents to only connect to a given site over HTTPS, even if the scheme chosen was HTTP. Browsers that have had HSTS set for a given site will transparently upgrade all requests to HTTPS. HSTS also tells the browser to treat TLS and certificate-related errors more strictly by disabling the ability for users to bypass the error page.

The header consists of one mandatory parameter (`max-age`) and two optional parameters (`includeSubDomains` and `preload`), separated by semicolons.

Directives

- `max-age`: how long user agents will redirect to HTTPS, in seconds
- `includeSubDomains`: whether user agents should upgrade requests on subdomains
- `preload`: whether the site should be included in the [HSTS preload list](#)

`max-age` must be set to a minimum of six months (15768000), but longer periods such as two years (63072000) are recommended. Note that once this value is set, the site must continue to support HTTPS until the expiry time has been reached.

`includeSubDomains` notifies the browser that all subdomains of the current origin should also be upgraded via HSTS. For example, setting `includeSubDomains` on `domain.mozilla.com` will also set it on `host1.domain.mozilla.com` and `host2.domain.mozilla.com`. Extreme care is needed when setting the `includeSubDomains` flag, as it could disable sites on subdomains that don't yet have HTTPS enabled.

`preload` allows the website to be included in the [HSTS preload list](#), upon submission. As a result, web browsers will do HTTPS upgrades to the site without ever having to receive the initial HSTS header. This prevents downgrade attacks upon first use and is recommended for all high risk websites. Note that being included in the HSTS preload list requires that `includeSubDomains` also be set.

Examples

```
# Only connect to this site via HTTPS for the two years (recommended)
Strict-Transport-Security: max-age=63072000
```

```
# Only connect to this site and subdomains via HTTPS for the next two years and also include in the preload list
Strict-Transport-Security: max-age=63072000; includeSubDomains; preload
```

See Also

- [MDN on HTTP Strict Transport Security](#)
- [RFC6797: HTTP Strict Transport Security \(HSTS\)](#)

- A link to Redhat's [HSTS configuration](#) page.

https://access.redhat.com/solutions/1220063

Apps Yahoo Weather Nagios - Linux Nagios - Errors Nagios - Warnings Nagios - Home Duke Mail vSphere Client MariaDB Docume... Nagios-Biology TTS Imported

Issue

- How to enable HTTP Strict Transport Security (HSTS) on Apache httpd

Resolution

- Verify that Apache httpd's Header module is enabled at `/etc/httpd/conf/httpd.conf` or `/etc/httpd/conf.modules.d/00-base.conf`

```
LoadModule headers_module modules/mod_headers.so
```
- Add the following configuration globally at `/etc/httpd/conf.d/ssl.conf` to `<VirtualHost *:443>` virtual host or for each SSL-enabled virtual host:


```
Header always set Strict-Transport-Security "max-age=63072000; includeSubDomains"
```
- Configure the non-encrypted virtual host to redirect all traffic to HTTPS by adding the following:

```
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteCond %{HTTPS} off
RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI}
</IfModule>
```
- If the changes above were made while Apache httpd was running, you will need to restart your Apache httpd server.

Product(s) [Red Hat JBoss Web Server](#) [Red Hat Enterprise Linux](#) Component [httpd](#) Category [Secure](#) Tags [http](#) [https](#) [security](#) [ssl](#)

This solution is part of Red Hat's fast-track publication program, providing a huge library of solutions that Red Hat engineers have created while supporting our customers. To give you the knowledge you need the instant it becomes available, these articles may be presented in a raw and unedited form.

Was this helpful?



Our DevOps Approach to Multisite and Drupal 8

A high-level overview

of how we're using DevOps to

pimp our stack

Kyle Skrinak, I.T. Manager

Web Development and Systems Administration

<https://kyle.skrinak.com>

(nom de cyber: @screenack)

Trinity College of Arts & Sciences at

Duke University

This presentation:

[2019-drupalcon-drupal-8-multisite.html](#)

Meet the team

- Two Drupal website designers and developers
- Two Symfony custom application developers
- Three sysadmins, including one Drupal lead
- Host 380 websites, including 108 Drupal websites
- Highly normalized platform — branding, features, and content
- Support third-party developed websites
- Plus a bevy of other web and sysadmin-related support
- We maintain an excellent support reputation

Current Infrastructure

- Duke OIT provided VM's
- RHEL7
- MySQL > MariaDB
- PHP's aggressive EoL change
- Shibboleth SSO
- Third-party integrations: e.g., Qualtrics forms and Box for storage
- 108 websites on a clustered server pair
- A varnish and a pound load balancer.
- A staging and production server per website

Our Drupal 7 workflow

- Ansible pulls Drupal core updates from d.o.
- Three git repos for code
 - Global
 1. Our **sites/all** (*contrib modules, themes, and libraries*) repo
 2. **Trinity-level theme** (*Omega-based*) repo at sites/all/themes
 - per website
 1. **Website repo** (*theme, modules, libraries*) in the `sites/<sitename>` directory
- `drush`, `git` and `ansible` to manage our stack cluster

What of Sleeping Dogs?

- Our mature workflow serves our customers and ourselves well...
- ...but it fails us, moving forward
- Shifting external requirements continue to mount
 - Web accessibility and responsive design
 - Our security posture
 - Integrate CI/CD and QA
 - Increasingly rapid stack updates

Enter DevOps

What is DevOps?

- To radically yet *seamlessly* change our workflow
- Rethink roles, rules, interactions, expectations
- Stay current with tool and paradigm shifts
- Learning from our peers and contributing back
- Maintain and improve our service reputation

Has DevOps Helped Us?

Yes.

*“Achieving extraordinary results
from ordinary people.”*

DevOps?

You're Soaking in It

- Bi-weekly DevOps meeting
 - New concerns and opportunities
 - New tools
 - Paradigm shifts
 - Changing individual roles
- Refactor our distribution and workflow
- Cross-discipline training
- Presentations on new proposals and tech

Why multisite?

- Misleading FUD on multisite
- Stable
- n-factor reduction in duplicate stack code
- Faster and inexpensive code updates

Preparing for Drupal 8

- Confirm continued D.O. support for multisite past 9.x
 - Yep: <https://www.drupal.org/project/drupal/issues/2306013>
- Confirm our understanding of Drupal 8 multisite
 - Yep: <https://kyle.skrinak.com/posts/2017-05-26-drupal-8-multisite-documentation/> **BUT**
- `composer` replaces `drush` for managing Drupal
- A “pure” composer-based workflow conflicts with
 - Managing the individual website instances
 - JavaScript library dependency resolution

How, then?

Distribution reference

- Maintain a “reference” D8 installation for managing core, based on the Drupal Project composer template
 - <https://github.com/drupal-composer/drupal-project>
- Run `composer update` against the reference
 - If there are updates, this will also update the `composer.lock` file
- Commit the `composer.lock` file updates to the core git repo

Core and Code Deployment

Switch from `ansible` to `GitLab CE CI/CD`

- The value of recent GitLab updates
- Trigger deployment updates from the GitLab UI.

Oops... NoOps!

Core and Code Deployment

Use a GitLab's CI/CD to

- `composer install` with the updated lock file to distributed D8 cores.
 - Far less expensive than `composer update` on all cores.
- Run QA tests as part of a build process
 - (Still in development)
- Consistently run deployment scripts
- Deploy website code to our staging and production servers

Core and Website Updates

Run a `pipeline` on targeted servers

- For all child sites:
 - Move website to maintenance mode
 - Database and code backup for all child websites
- Pull composer.lock file
- Composer install (core only)
- For all child sites:
 - Drupal database update
 - Cache rebuild
- Reapply permissions on core or website directories only

GitLab CI/CD Demo

[Link](#)

Features and Benefits

- Eliminates need for manual deployment
- “True” push-button deployment for our devs
- Consistent deployment execution
- Consistent state restoration
- Easy access to deployment logs by all

Hindsight = 20/20

- Base our distribution on Acquia's Lightning distribution
 - <https://github.com/acquia/lightning>
- Build an installation profile

What's Next?

- Library management: big hairy eyeball
- Implement varnish and pound for D8
- Build and maintain our own installation profile
- Trigger deployments with commit tags
- Improve integration of website-level configuration management
- Backport what we can to our D7 workflow

Questions?

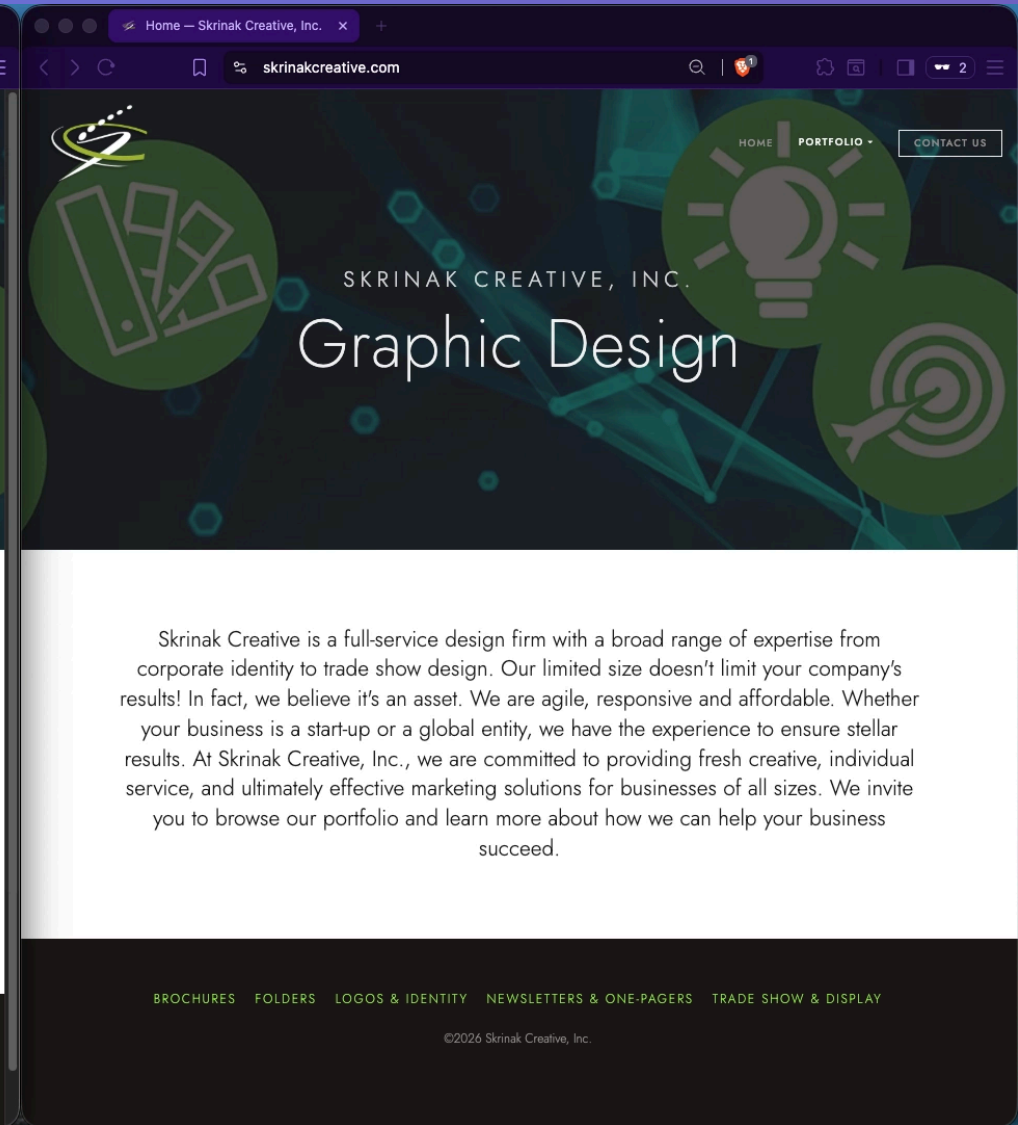
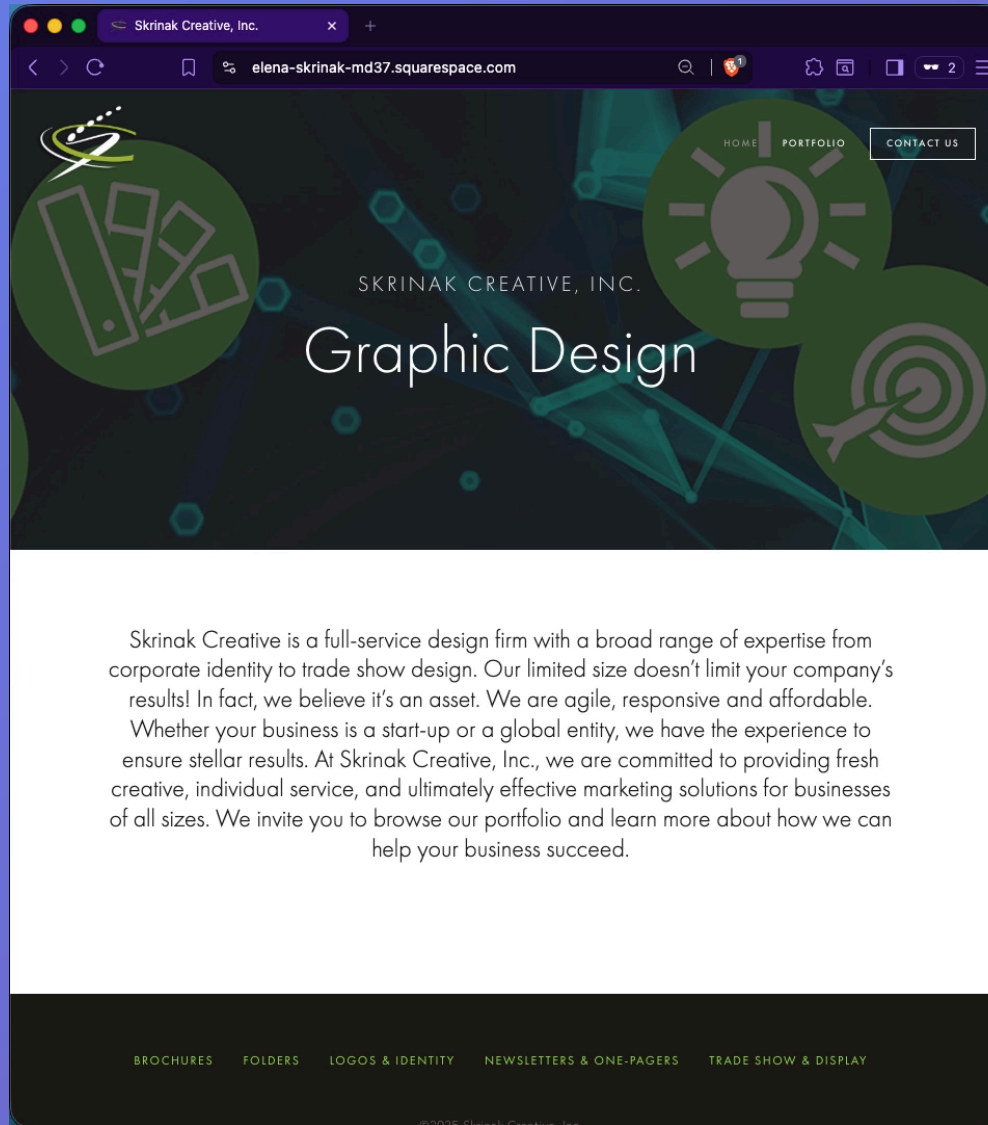
?

Thank You!

!

AI-Accelerated CMS Migration with Claude Code

From Squarespace to Astro in eight hours



original site — migrated site

The Situation

My wife Elena runs a graphic design business — skrinakcreative.com

- On Squarespace since 2017
- Squarespace worked well
- Her business downsized
- Seven pages, portfolio galleries, minimal interactivity
- **\$16–23/month → \$0**

What I Considered First — HTTrack

Web scraping is the obvious starting point. I know HTTrack well.

- Used HTTrack + AI as a first attempt
- Results were unusable
- Platform-coupled assets would require significant decoupling effort
- Unclear IP standing on scraped theme assets
- Enough friction to make me curious about a cleaner approach

The Constraint — Know Your Scope

This was a **simple** migration. Knowing that upfront matters.

- 7 pages, single grid-based theme
- One integration swap: Google Maps → OpenStreetMap
- No forms, no authentication, no e-commerce
- No CMS data model to untangle

The Approach — Discipline Before Code

Classic project management applied to an AI-accelerated workflow.

Before any code was written:

1. **Plan** — goals ranked by priority, options evaluated
2. **Scope** — seven pages, exact asset inventory, known integrations
3. **Specification** — design tokens, typography, color palette, spacing
4. **Testing requirements** — visual regression baselines, pass criteria defined

The Stack Decision

Astro + Cloudflare Pages

Decision	Choice	Rationale
Hosting	Cloudflare Pages	Free tier, global CDN, auto-deploy from git
Framework	Astro	Native image optimization, clean component model
Theme	None — built from scratch	Existing themes didn't match Elena's design
Font	Jost (Google Fonts)	Adobe Fonts license lost on Squarespace cancellation
Lightbox	GLightbox	Matched portfolio gallery behavior exactly

What Claude Code Actually Did

Claude Code wrote **all of it**. My role was project management.

- Discovery & audit, technology evaluation, design spec extraction
- Astro scaffold, component build, gallery implementation
- Visual alignment, test suite, deployment configuration

The standout: Layout identification and visual alignment

Describing design corrections in plain English, with a visual comparison test suite confirming each change against the live Squarespace baseline.

The Test Suite

78 tests. Claude Code wrote them.

- **60 visual regression tests** — 7 pages × 3 breakpoints × 2 test types
- **18 navigation tests** — desktop and mobile, dropdown and drawer behavior
- Playwright baselines pulled from the live Squarespace site

I already take parallel test suite management for granted. That shift happened fast.

The Numbers

Total time: 8 hours — start to deployed.

- ~1 hour lost to a Cloudflare DNS configuration miss
- A clean run: probably **7 hours**

Metric	Squarespace	Astro	Δ
Performance	57	86	+29
Accessibility	96	99	+3
Best Practices	100	100	—
SEO	93	92	-1

The Results

Deployed. Stakeholder approved.

- Elena signed off — she cares about the design, not the tech
- Not pixel-perfect: font substitution accounts for most of the gap
- Better on every meaningful performance dimension
- Full control, no vendor lock-in

Project Scope

This was the simple case. Extrapolate carefully.

- 7 pages, one theme, one clean integration swap
- Complexity scales with forms, data models, authentication
- AI instructions fail — verify infrastructure configuration yourself

Prior assumptions about effort may no longer hold. Test it yourself.

The Takeaway

Context is the input. Quality is the output.

- Define scope before you start
- Write specs before you build
- Require tests before you ship
- Manage AI like a very fast, very literal contractor

The question isn't whether AI can do this. It's whether *you* can manage it well enough to let it.

Questions?

See it live

- skrinakcreative.com — production site, now running on Astro
- elena-skrinak-md37.squarespace.com — original Squarespace site, available until September 12, 2026

Read the full story

- [From Squarespace to Astro: A Migration Story](#) — full narrative with project context, decision-making process, and lessons learned

The stack

- [Astro](#) · [Cloudflare Pages](#) · [GLightbox](#)
- [Claude Code](#)

Duke University's

Code+

co-curricular program

promoting undergraduate

Diversity / Equity / Inclusion

Link to presentation: [../posts/2022-04-07-code-presentation/](https://www.linkedin.com/posts/2022-04-07-code-presentation/)

Kyle Skrinak

<https://www.drupal.org/u/screenack>

Senior Manager, I.T., Duke Web Services Development and Support

Duke University

<https://kyle.skrinak.com>

Our team builds, hosts, and maintains over 250 Drupal websites, including multisite, headless, and other websites and applications.

What is Code+?

8/23/18, 3:00 PM

App Demo

...

The screenshot shows a mobile application interface. At the top, the status bar displays the time 9:41, signal strength, and battery level. Below the status bar is a navigation bar with a back arrow and the text "Select Destination". The main content area shows "Duke Cancer Center" with a right-pointing arrow icon. Below this, the date and time "8/23/18, 3:00 PM" are displayed. A map of the Duke University campus is visible in the background, with a white date selection menu overlaid on it. The menu lists dates from Tuesday, August 21 to Saturday, August 25, with corresponding times. A hand cursor is pointing at the entry for Thursday, August 23, at 3:00 PM. An "OK" button is located at the bottom of the menu.

Day	Time
Tue Aug 21	1:00
Wed Aug 22	2:30 AM
Thu Aug 23	3:00 PM
Fri Aug 24	4:30
Sat Aug 25	5:00
Sun Aug 26	6:00

Origins for the Drupal module project

- A re-org at Duke
- Office of Information Technology's call for Code+ projects
- A mutually-beneficial project
 - Supports the Code+ team, Drupal, Open Source, and Duke
- Duke maintains an internally-developed module on D.O. in need of attention
 - AKA "BasicShib" <https://www.drupal.org/project/basicshib>
- A Duke need: integrate Grouper with Shibboleth

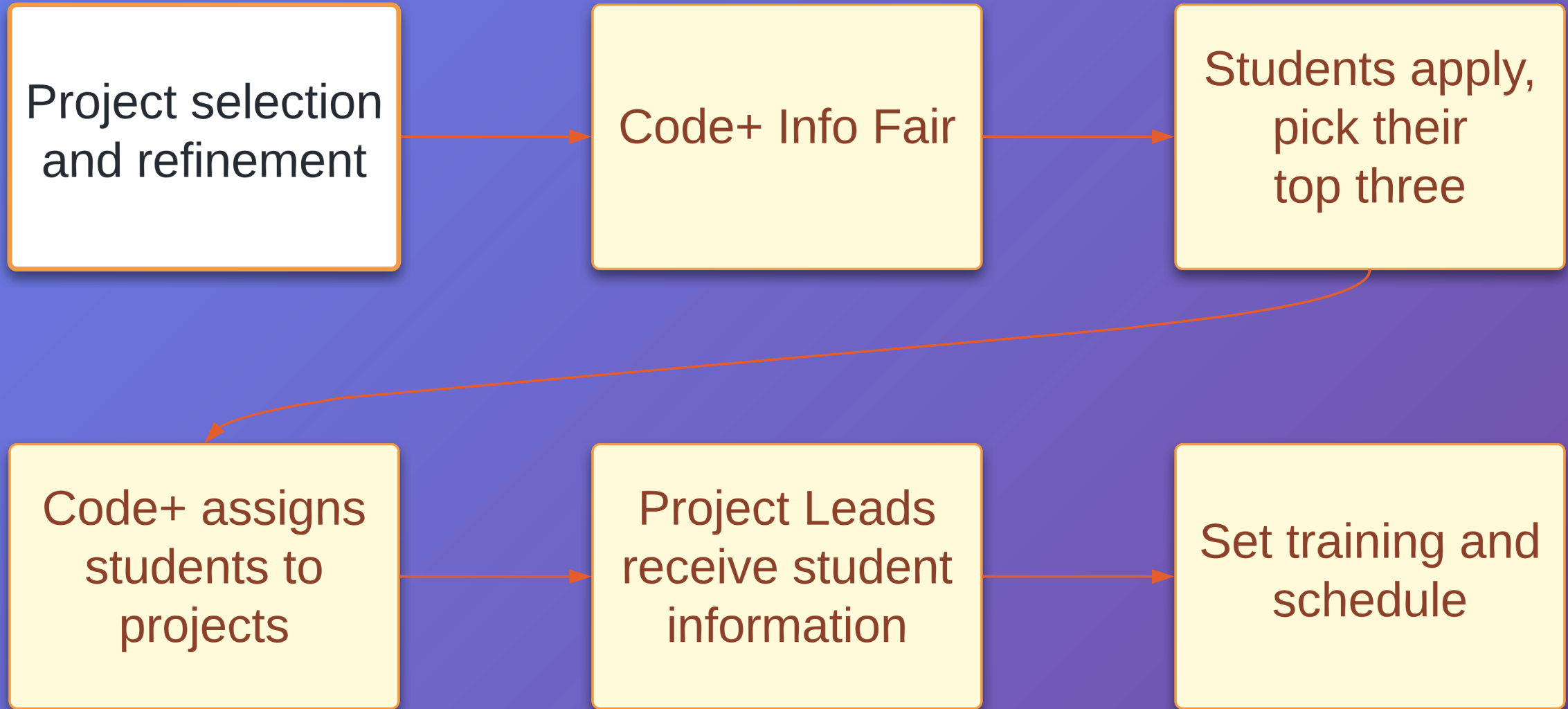
A Code+ Project is born

A Code+ Project is born

Drupal Module for Software Design and Development

The Drupal open-source software platform provides an opportunity for those who wish to design, develop, release, and finally maintain a non-commercial software project. A team of students will work with Duke's Office of Information Technology and other partners across campus to develop a Drupal module to extend the functionality of the existing framework to create an application. Students will work with Drupal 9 to extend Symfony's PHP object model paradigm. This allows students to work within a fully compliant Object-Oriented Programming, or OOP, paradigm.

A Code+ Project is Born



A Code+ Project is Born

Project selection
and refinement

Code+ Info Fair

Students apply,
pick their
top three

Code+ assigns
students to
projects

Project Leads
receive student
information

Set training and
schedule

A Code+ Project is Born

Project selection
and refinement

Code+ Info Fair

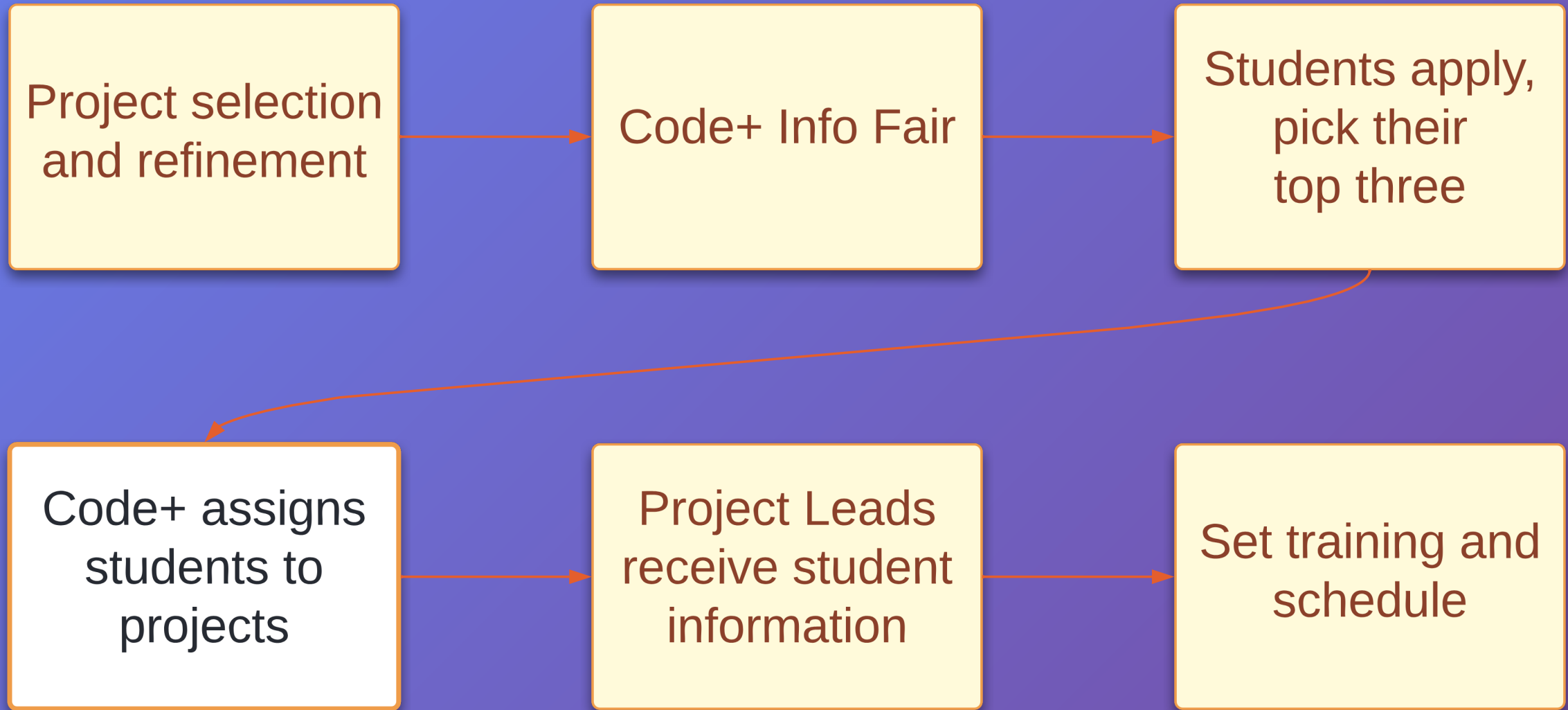
Students apply,
pick their
top three

Code+ assigns
students to
projects

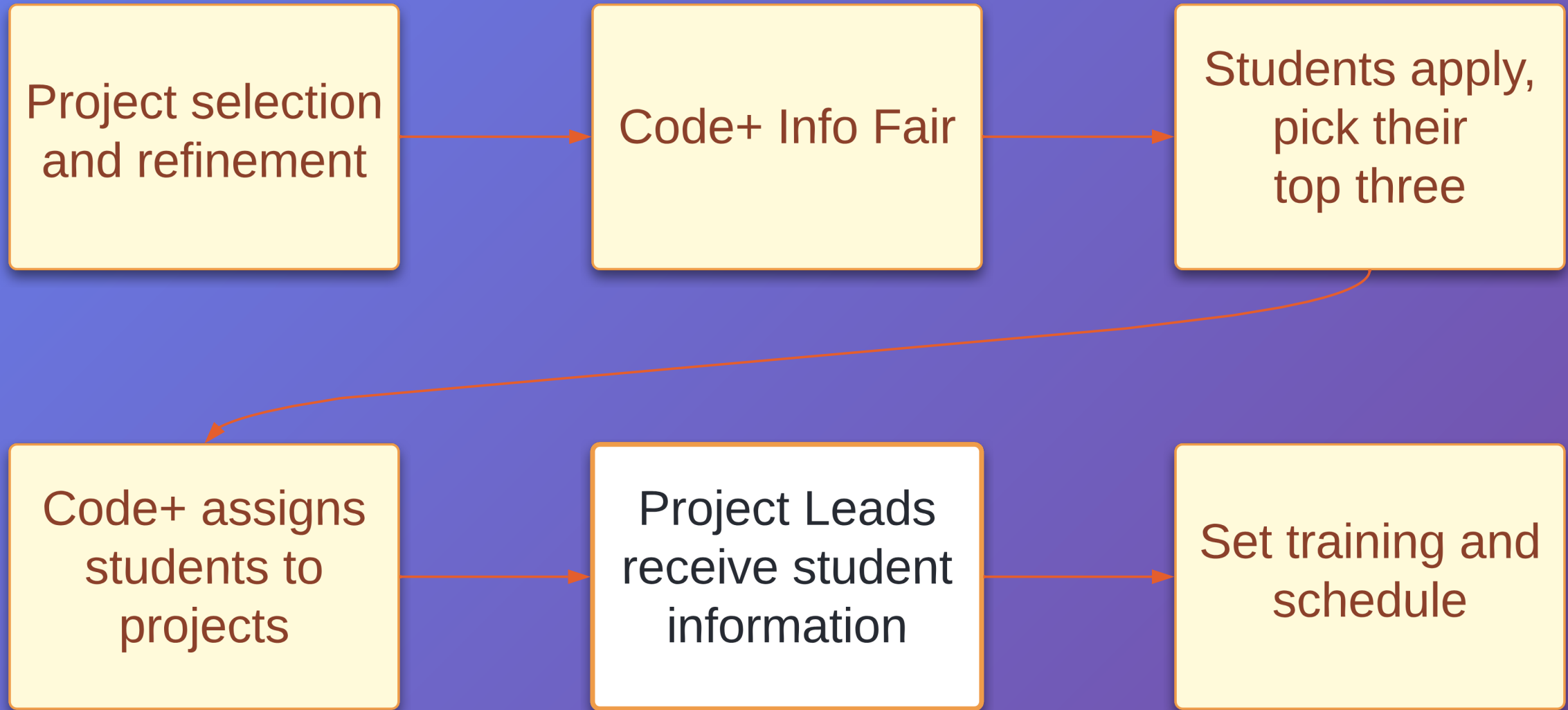
Project Leads
receive student
information

Set training and
schedule

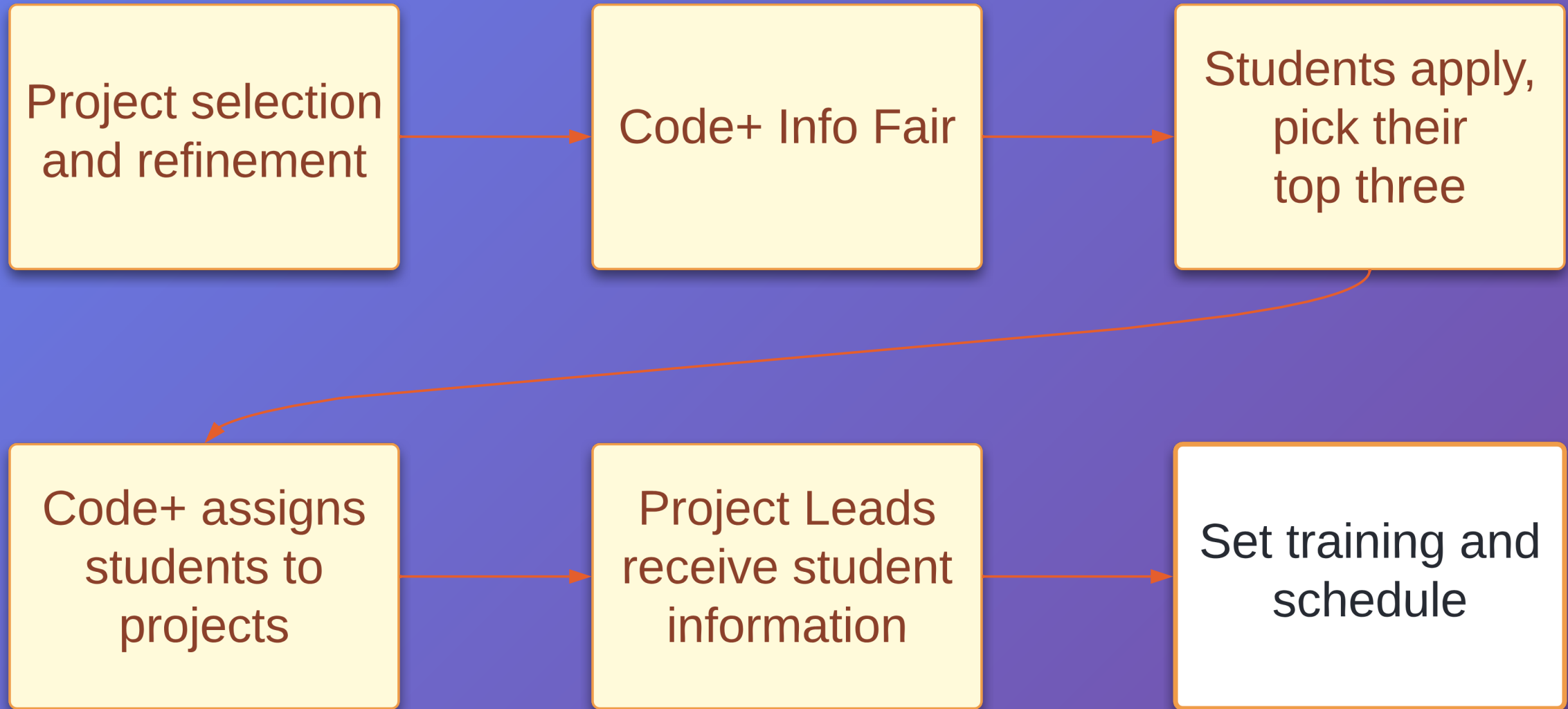
A Code+ Project is Born



A Code+ Project is Born



A Code+ Project is Born



Project Parameters

- Code+ administrators assigns five students to project
- We discuss project, training, and software requirements
 - Lando for development.
 - Identify common means for collaboration.
- Daily check-in and check-out on Microsoft Teams and Zoom
 - We had a mix of on-campus and remote students across time zones.
 - Encourage team collaboration on a standard platform.

Project Parameters

Project Parameters

Project Successes

Project Outcomes

Challenges, Learnings, and Next Steps

Challenges, Learnings, and Next Steps

Questions?

Introduction to Drupal

This presentation:

[drupal-intro.html](#)

Your presenter

Kyle Skrinak, I.T. Manager

Web Development and Systems Administration

<https://kyle.skrinak.com>

Trinity College of Arts & Sciences at

Duke University

What we're covering today

- Editing content at TCA&S using Drupal
- Managing department profile data
- What is web accessibility and why should I care?
- Getting support

Who is this training for?

- It's your first time editing a TCA&S website
- You're new to Duke or Trinity
- You've edited WordPress in the past
- You don't know HTML from a ham sandwich

Editing content at TCA&S using Drupal

<https://admin.trinity.duke.edu/technology/drupal-website-editing>

Managing department profile data

[tts-profile-mgmt.html](#)

Web Accessibility

A video presentation on
“What is Web Accessibility?”

[Link to accessibility video](#)

How does this pertain to me?

For more info at Duke, check out

[Duke's Accesibility website link](#)

Getting support

trinitywebsupport@duke.edu

And not any single TTS staff member's address, please.

You can find online documentation at

<https://bit.ly/TTS-Drupal-Documentation>

How'd I do?

https://duke.qualtrics.com/jfe/form/SV_7ZYqxa5Jn6debo9 (link no longer accessible)

Questions?

?

Thank You!

!

Drupal Multisite on a Dime

or How we Adopted CI/CD for TCA&S's Drupal Multisite Hosting for Trinity's 100+ Drupal websites

This presentation:

[../posts/2020-12-03-drupal-multisite-on-a-dime/](#)

Your presenter

Kyle Skrinak, I.T. Manager

Trinity College of Arts & Sciences

Web Support (now part of OIT), Duke University <https://kyle.skrinak.com>

Meet the team

- David Palmer, *DevOps*
- Andy Smith, *Web Dev & Design*
- Gabe Fahl, *PHP Programmer*
- Alex Verhoeven, *Web Dev & Design*
- John Herr, *PHP Programmer*

Who We Serve?

- Trinity College of Arts & Sciences departments, administrative, labs, etc.
 - 120 active websites
 - 100+ legacy websites and web applications
- SLAs and MoUs, *including*
 - divinity.duke.edu
 - dibs.duke.edu
 - sanford.duke.edu

Requirements

- Consistency across all of our websites
- Rely on Open Source community, and not custom module development
- Decreased support and discovery time
- Greatly agility with code updates, website maintenance, and deployment
- Minimize cost for external tools

Overview: Timeline

- 2013: Apostrophe → Drupal migration starts
- 2014: Drupal Multi-site (*hosting, version control*)
- 2015: Base + Sub theme (*branding, gitorius → gitlab*)
- 2016: FDS → Scholars migration (*profile data, membership*)
- 2017: Drupal 8 distribution begins, first D8 website (A → D7 migration ends)
- 2018: D8 requirements set (*hosting, branding, membership*)
- 2019: D8 distribution starts, production begins (CI/CD deployed)
- 2020: Currently at 27 D8 websites

Innovations

- 2014: Drupal Multi-site
 - Pantheon discussions and debate over multi-site
 - The DevOps-like dashboard for managing websites
- 2015: Drupal Omega Base + Sub theme
 - Simplified enforcing our branding across n number of websites
 - Reduced complexity of our support by providing reasonable defaults for content migration

Innovations

- 2016: Department profile management
 - The “People” section of our websites
 - Faculty Database System (FDS) → Scholars effort
 - Missing: non-faculty membership and our TCA&S-wide directory on trinity.duke.edu
 - Support for Duke’s system of record
 - Solution: design and implement middleware
 - Directory Tool
 - Plenty of Feeds

Drupal 8: Design phase

- Finished D7 website migration by 2017
- Goal setting
 - Enforce feature and content denormalization
 - Continue to lean on community, not custom modules
 - Confirm: multi-site
 - Configuration management
 - Exploit DevOps tools: Composer, CI/CD

DevOps becomes a real thing

- You've got developers in my sysadmins?
- You've got sysadmins in my developers?

Mmmm! It's good!

Our CI/CD Pipeline

Site tasks	Pushbuttons	Revert	Migrate
Enable maintenance mode	Production backsync	Restore DB	Staging to Prod
DB Backup	Git pull <branch>	Revert code	
Git Reset --hard <branch>			
Drush entity update			
Drush clear caches			
Apache OpCache clear			
Drush config export			
Drush config import			
Fix permissions			
Disable maintenance mode			
Backsync DB from prod			

Our Current Workflow

- Continue to enhance our content editorial experience
- Weekly meetings with the TCA&S Comm team
- Tamp down scope creep as we scale
- Synchronize core features and display across all websites
- Extend our CI/CD to WordPress and other servers
- In process: default site content

Q & A

thank you

How to Manage Departmental Faculty and Staff Data

This presentation:

[tts-profile-mgmt.html](#)

Kyle Skrinak, I.T. Manager

Web Development and Systems Administration

<https://kyle.skrinak.com>

Trinity College of Arts & Sciences (TCA&S), *now part of OIT*,

Duke University

This presentation:

[tts-profile-mgmt.html](#)

What we're covering today

Updating:

- Updating faculty and staff profile data
- Department group membership not in Scholars
- Assigning administrative assignments

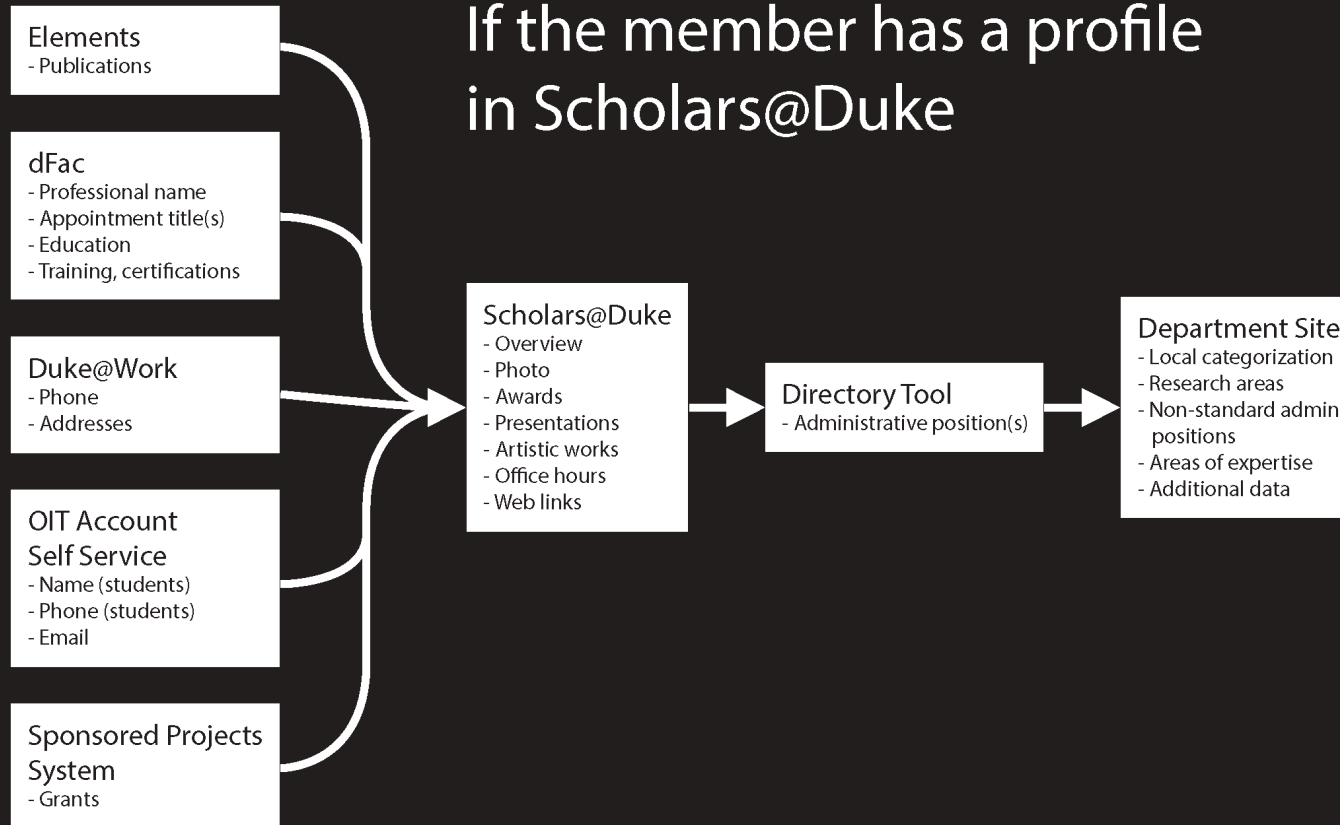
Who Does What?

- *Content editors* are responsible for the content of their website.
- Your *power users* are responsible for updating profile information
 - Any department member may update his or her profile as well
 - The Scholars list of power users is here: <https://about.scholars.duke.edu/locate-power-user>
- The *TCA&S team* is responsible for the reliable operation of our directory
- The *Scholars team* manages the Scholars web site and data interaction

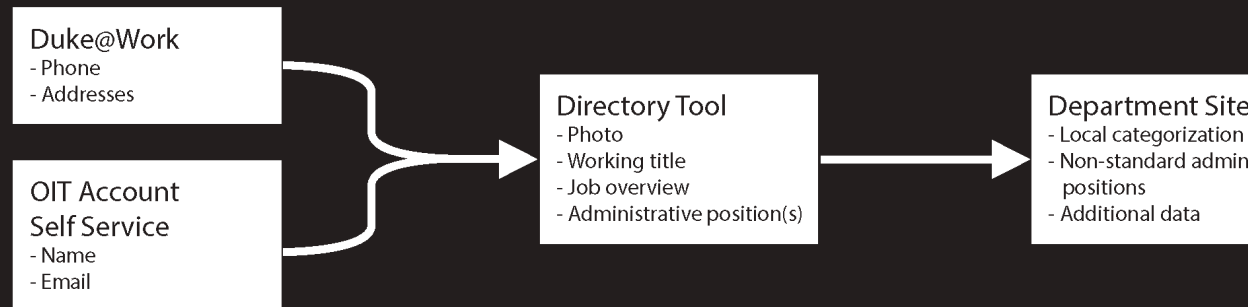
Where to Make Profile Updates?

- We support Duke's "System of Record."
- <https://directorytool.trinity.duke.edu>
 - <https://directorytool.trinity.duke.edu/help/profile-data-source>
Handy reference
- scholars.duke.edu
- Your department's website for any unique profile data
- The next two slides provide a different view of the data sources

If the member has a profile in Scholars@Duke



If the member does not have a profile in Scholars@Duke



Remember

1. Your group's profile data also goes to trinity.duke.edu's master directory for all Trinity College of Arts & Sciences members.

<https://trinity.duke.edu/directory>

2. You may have to wait up to 48 hours to see your profile changes apply to your web site.
3. What about unpublished department member nodes?

Who Do I Contact for Help?

- **Simple rule:** Confirm Scholars profile data. If Scholars data is incorrect, then contact your power user.
- If Scholars profile data is correct, but your website's data is incorrect, then contact:

trinitywebsupport@duke.edu

And not any single TCA&S staff member's address, please.

Questions?

?

Thank You!

!

What I did

at the 2022 DrupalCon,
Portland, OR

My summary

Trimming the fat

