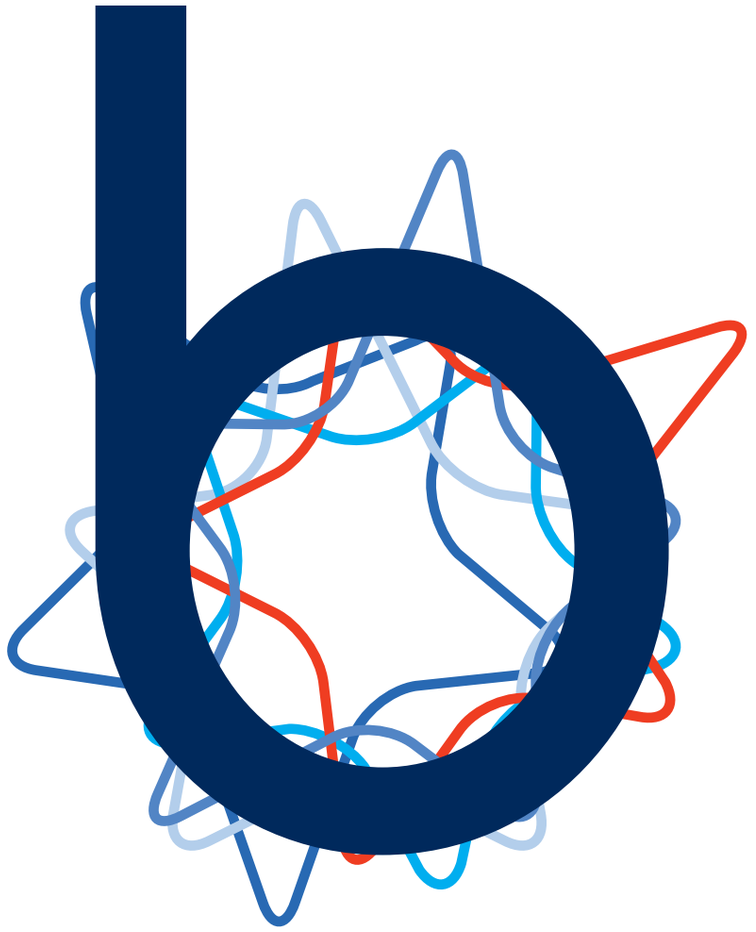
Best practices for the development and deployment of robust Drupal applications

Adrian Rollett [bluespark / roomify.us]

Technical Director / Bluespark

full-service web agency
long record of contribution
design . ux . development



bluespark

Co-founder /
roomify.us

roomify

product spin-off from bluespark
flexible booking solutions
drupal-based . open-source

Goals

- build (and build upon!) robust Drupal applications
- learn useful processes / philosophies
- discuss implementation strategies

Dean Hochman . Flickr

What makes an application a good application?

“any program is only as good as it is useful.”

–Linus Torvalds

- does what it’s asked
- performs well
- stable and reliable
- functional and attractive on *all* devices

Preparation

What is your process?

proc·ess

noun

a series of actions or steps taken in order to achieve a particular end.

What are the elements of **your** process?

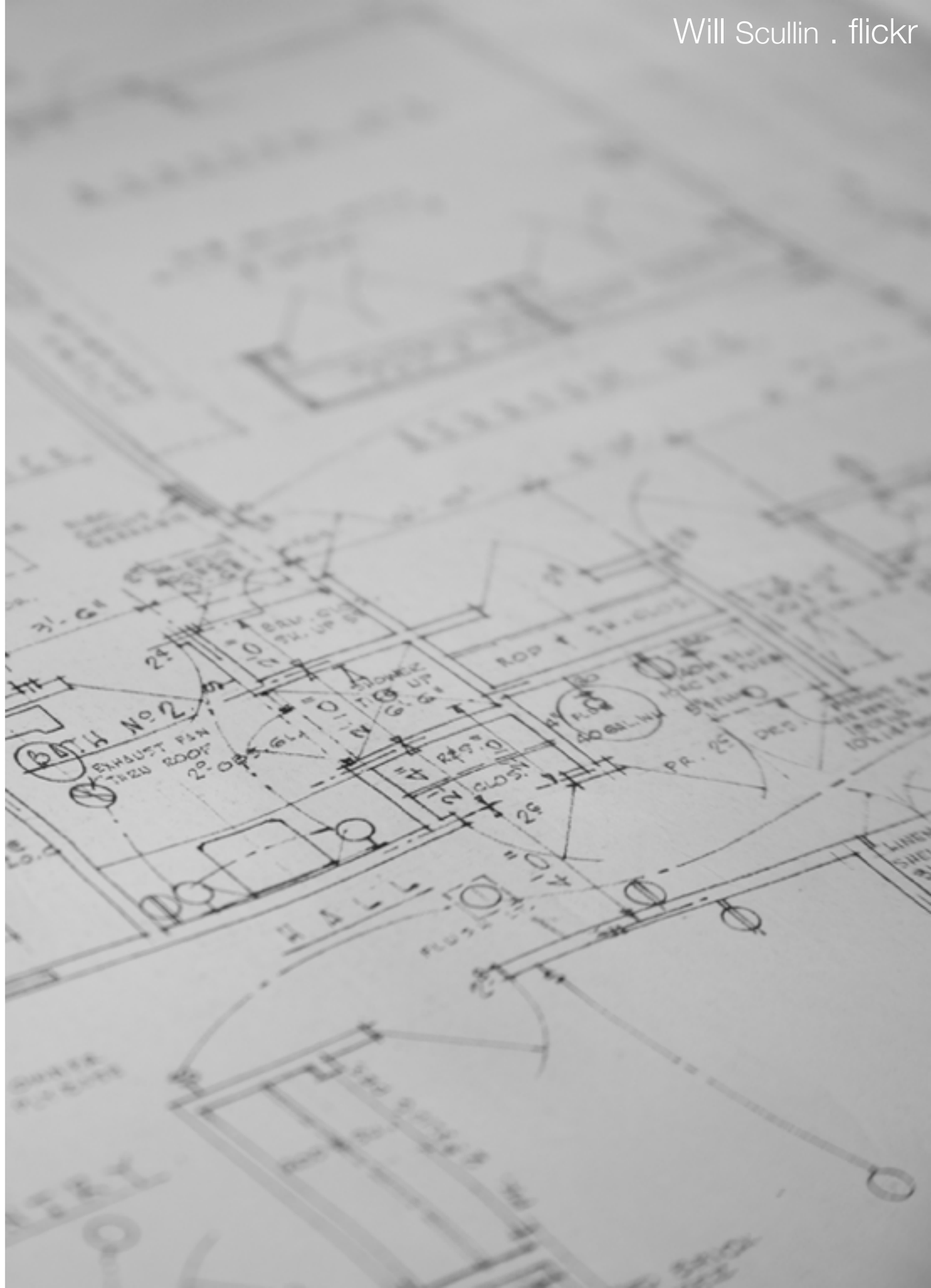
methodology . patterns . best practices

the Order of Things

- user experience design
- visual design
- technical design

Technical Design

- best done with multiple people: we involve members of the UX, design and technical teams
- time-consuming but worthwhile
- plan and estimate
- finished production: complete implementation timeline + tickets



Working within Budget

- If the client is completely bought into an agile process, budget per sprint and iterate
- If project is fixed budget, up front planning and communication of scope is completely essential

(from apelbaum.wordpress.com) [1]

The Build-out

There is no substitute for proper technical planning

Managing time during Development

- Agile
- Kan-ban
- Waterfall

How we do it

- Agile sprints with explicit QA/Code review steps for individual tickets
- Managed within the overall scope/timeline framework defined during the Technical design and estimation process

Parallel development / External teams

- Plan first, then execute
- Dependency order is vitally important to figure out up front
- Specify, specify, specify

QA

Bill Sempf
@sempf

Follow

QA Engineer walks into a bar. Orders a beer. Orders 0 beers. Orders 999999999 beers. Orders a lizard. Orders -1 beers. Orders a sfdeljknesv.

RETWEETS 17,418 FAVORITES 10,002

10:56 AM - 23 Sep 2014

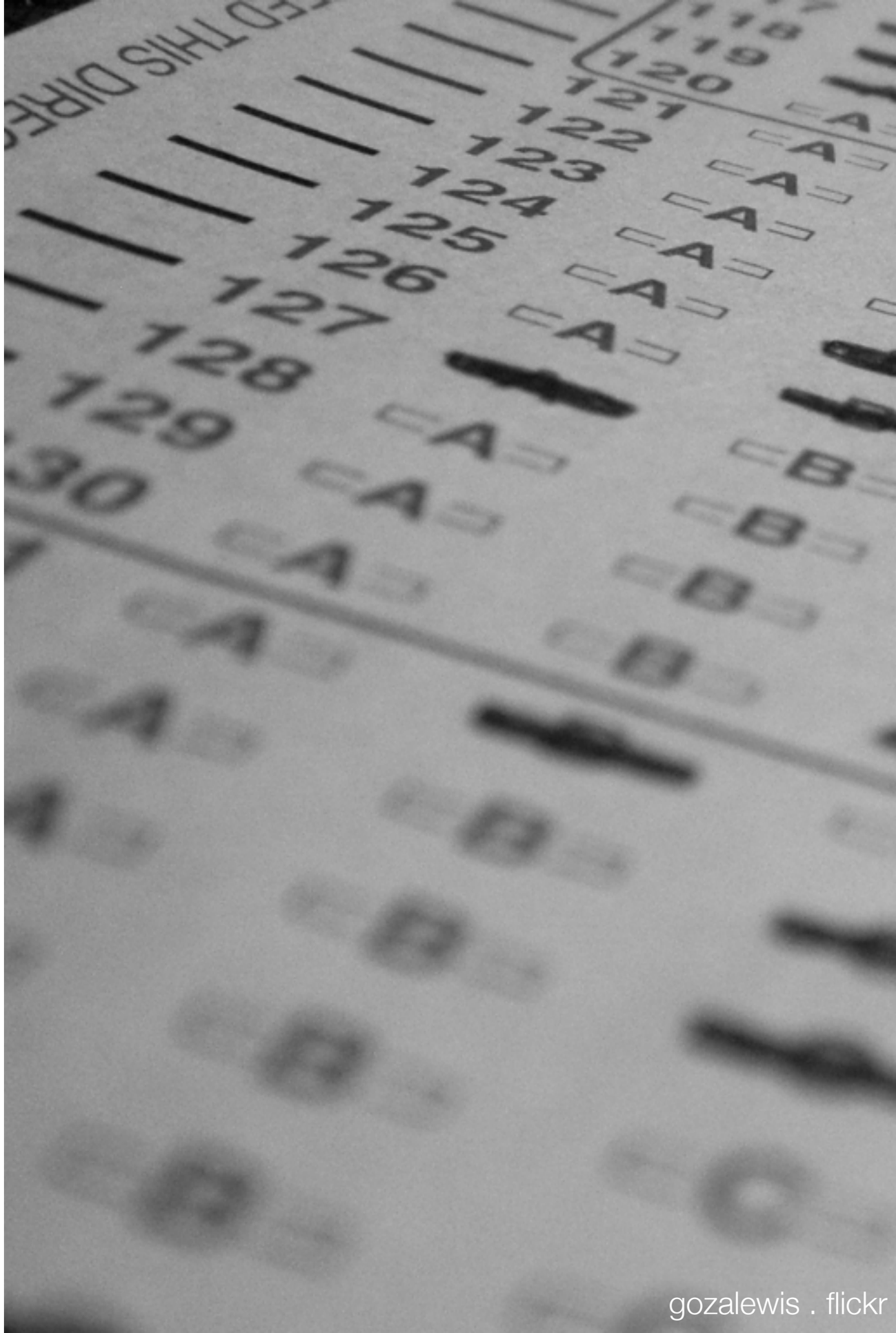
“QA is culture, not a step.”

Where does QA happen?

- Up front, during technical design
- During development - code standards matter!
- Code review
- Automated code testing
- Formal QA

Testing

- The two most important things you can know about testing:
  - Do it!
  - Automate it - tests that don’t get run automatically don’t do their job
- For more than you wanted to know, see our presentation: Quantifying the Value of Devops [3]



Just Add Light . flickr

On-going development - balancing concerns

Define the goals

- new features
- bug fixes
- continued stability

How do we meet them?

“Undeployed Code Is Inventory;
Inventory Is Risk.”

–xaprb.com [2]

The more you deploy, the easier it gets

but don’t deployments take lots of time?

Automate, automate, automate

- Work towards a model where each functional pice of code is tested individually and **automatically** in its own environment

- Make sure the results are seen by humans - we run tests via jenkins and send the output to our company chat

- This environment should be identical to the production environment in all aspects possible

- You can build it, or you can buy it

Testing

- Any test is better than none, **IF** it runs automatically
- Don’t let perfect be the enemy of good, it’s OK not to have 100% test coverage
- Identify what must work, and test that functionality
  - ecommerce: cart/checkout
  - brochure site: home page, contact form
  - intranet: user login, permissions
  - and so on

Extra Credit: Continuous Deployment

- Ensure all deployment steps exist in code that is itself reviewed and tested
- Don’t deploy new code without new tests
- All tests must pass to deploy
- Rollback must also be automated

Conclusions

- Define your process and make your workflows fit it
- Plan *before* you build
- Automate all the things
- Test the important things
- Deploy as frequently as possible

seattle municipal archives . flickr

Questions?

Thank you!

adrian@roomify.us
twitter.com/acrollet

Resources

- [1] Choosing Between Schedule, Budget, Scope, and Quality

- [2] Code Freezes don’t prevent outages

- [3] Quantifying the business value of devops