Team Driven Development

Daniel A. Lewis

Web Applications Team Lead | Marketing Technology

Introduction

Who am I?

My Background

- Avid software developer from age 9
- Member and leader of several development user groups
- Previous roles
  - Earthlink Support Engineer
  - Helpdesk and Solutions Engineer for a small plastics company
  - Network Engineer for an IT Services Organization
  - Self Made Developer at said IT Services Organization
  - Senior Developer for an Automotive CRM company
  - Web Applications Developer at Esri

Why I Like Writing Software

What it means to me to be a carpenter of the web

Why I like Doing It Alone

The story of NetOps

Or how I convinced someone to pay me to write software

Why I Don’t like it when others build software by themselves

The story of the 3000 line Sproc

Or how I came to hate SQL

Exploring the Process of Building Software

We’ve done this a few times now – it should be getting easier

All the wonderful development philosophies

- Waterfall
- Scrum
- Agile
- WAgile
- Extreme Programming
- Kanban

- Or my favorite:
- Post-it notes, Pizza, and a problem.

Getting the Team Aligned

We don’t have to get on the same page,
but let’s at least try to use the same book.

Getting all the great minds together to work on the same problem at the same time

Take roll

Learn the strengths and weaknesses of each team member

Never assume anything, ASK!

Let the team define the playbook

- Agree on a tech stack and how new technologies will be evaluated.
- Let the team establish a coding standard, and other working parameters.

Something to remember

Make the team a safe place

Taking on new projects

Did you groom the backlog with your epic story about that sprint where you tripped and scrummed up your knee?

What’s working for us

- Project is presented to the team lead who briefs the team
- Team discusses, comes up with ideas and questions
- Team decides who is going to work on what parts
- Members are paired up on each responsibility
- Project owner and other stake holders are invited to meet with the team to kick things off

Doing the Work

The art of getting things done.

Some of our processes

- Test Driven Development…
  - Red, Green, Refactor, Test, is closer to the truth.
- Team Driven Development
  - Pair, Tri, and even Quad programming at times
  - Asking each other for help
  - Trying things we don’t know how to do
  - Experimenting as a team
  - Testing each other’s work
  - Questioning each other’s decisions
- Asking why a lot

- Building with our stakeholders, not for them.

Quality Assurance as a team

We have no QA Department (and that’s a good thing)

Some of our QA processes

- Once code runs on one machine test it on another
- Deploy to testing servers often
- Show work to stake holders in increments
- Spend dev time with designers
- Test on multiple browsers and devices regularly
- Automation (unit tests, selenium tests)
- Team Code Reviews (before anything goes into production)
- Daily QA Demo / check-ins with lead
- Full project team reviews before launch

Improving our Skills

We are committed to learning and keeping our swords sharp

Our commitment to learning

- Sharing knowledge through the process of building software
- Demoing code and doing code reviews
- Setting aside time for self-improvement
- Trying new technologies on small projects
- Taking lead on areas not strong in
- Focus on learning non-technical skills

Nurturing the Team

Things I’ve found that work.

Retrospectives

Talk honest and openly about what’s working and what’s not

Allow all things to be subject to change

Retrospectives

Always focus on the positive

Don’t take all the credit, share the love

Retrospectives

Never ignore the negative

Don’t let it be someone else’s fault

Let the team make as many decisions as possible

Retrospectives

Evaluate individual performance using team accomplishments

Retrospectives

Don’t dictate process, ask for results

Closing Thoughts

Let’s reflect on this reflection.

The things I’d like you to take away from this talk

- People like interacting with others (even software developers)
- Building software should be fun
- There is no one right way to build software, but many proven ways to fail
- By letting the team self-manage interesting things happen
- Leading a team is about setting them up for success, not telling them how to succeed.


esri
THE
SCIENCE
OF
WHERE

Aspect Ratio Test