

# Agile in the Enterprise

## Nature Network Case Study

**Matt Jankowski**  
**[mjankowski@thoughtbot.com](mailto:mjankowski@thoughtbot.com)**

**<http://www.thoughtbot.com/>**  
**<http://giantrobots.thoughtbot.com/>**



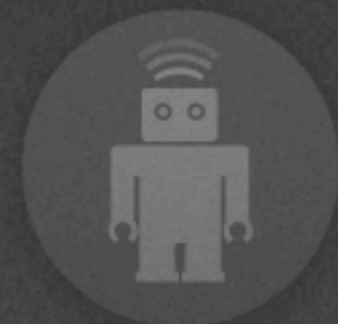
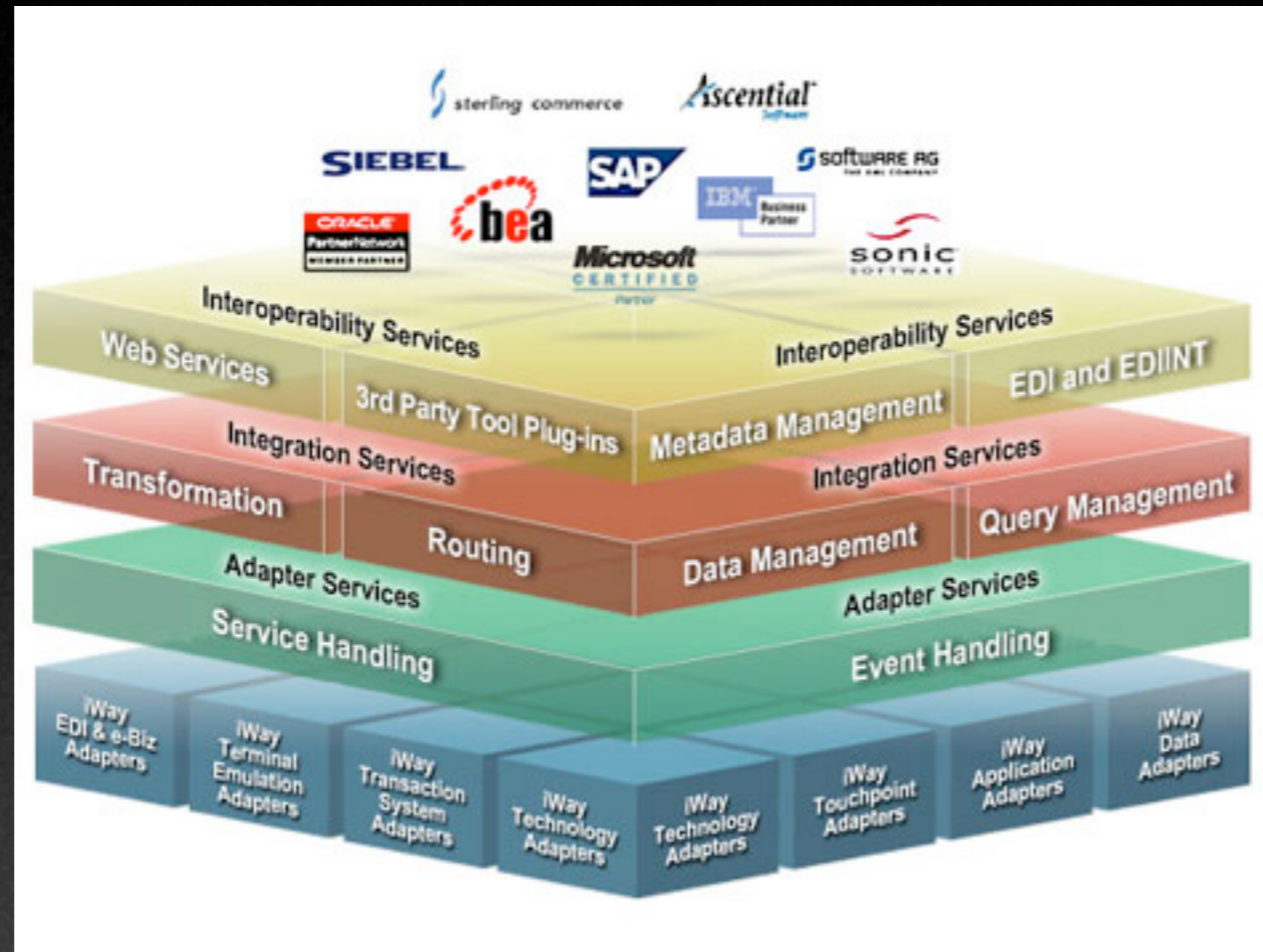
# Agile IN the enterprise?

...

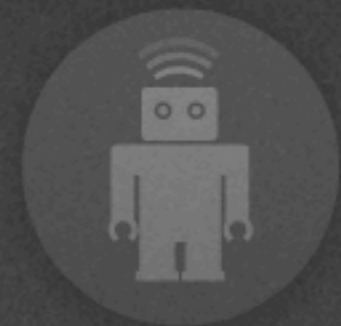
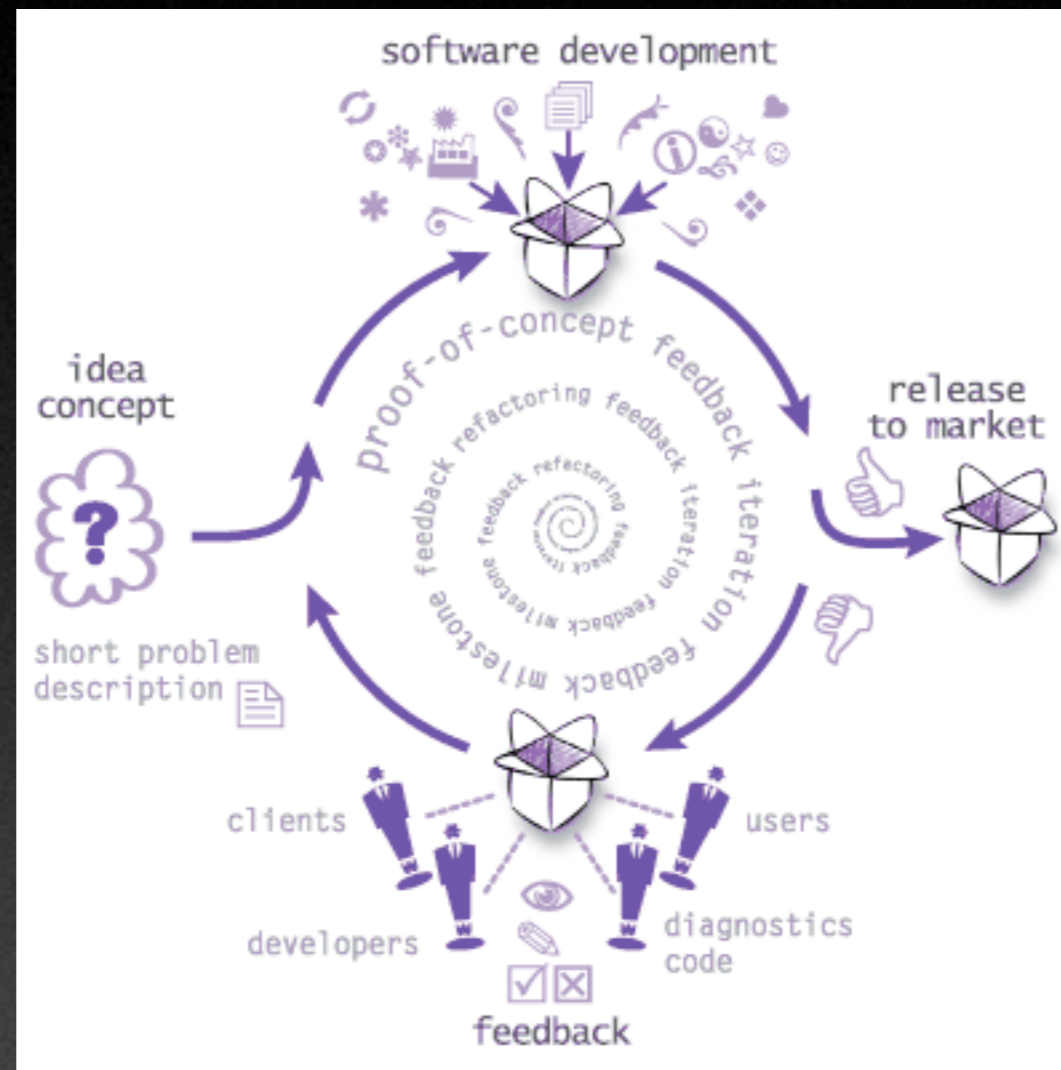
# Agile VS the enterprise?



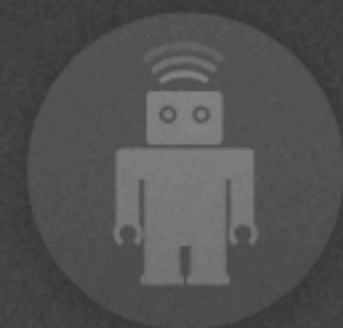
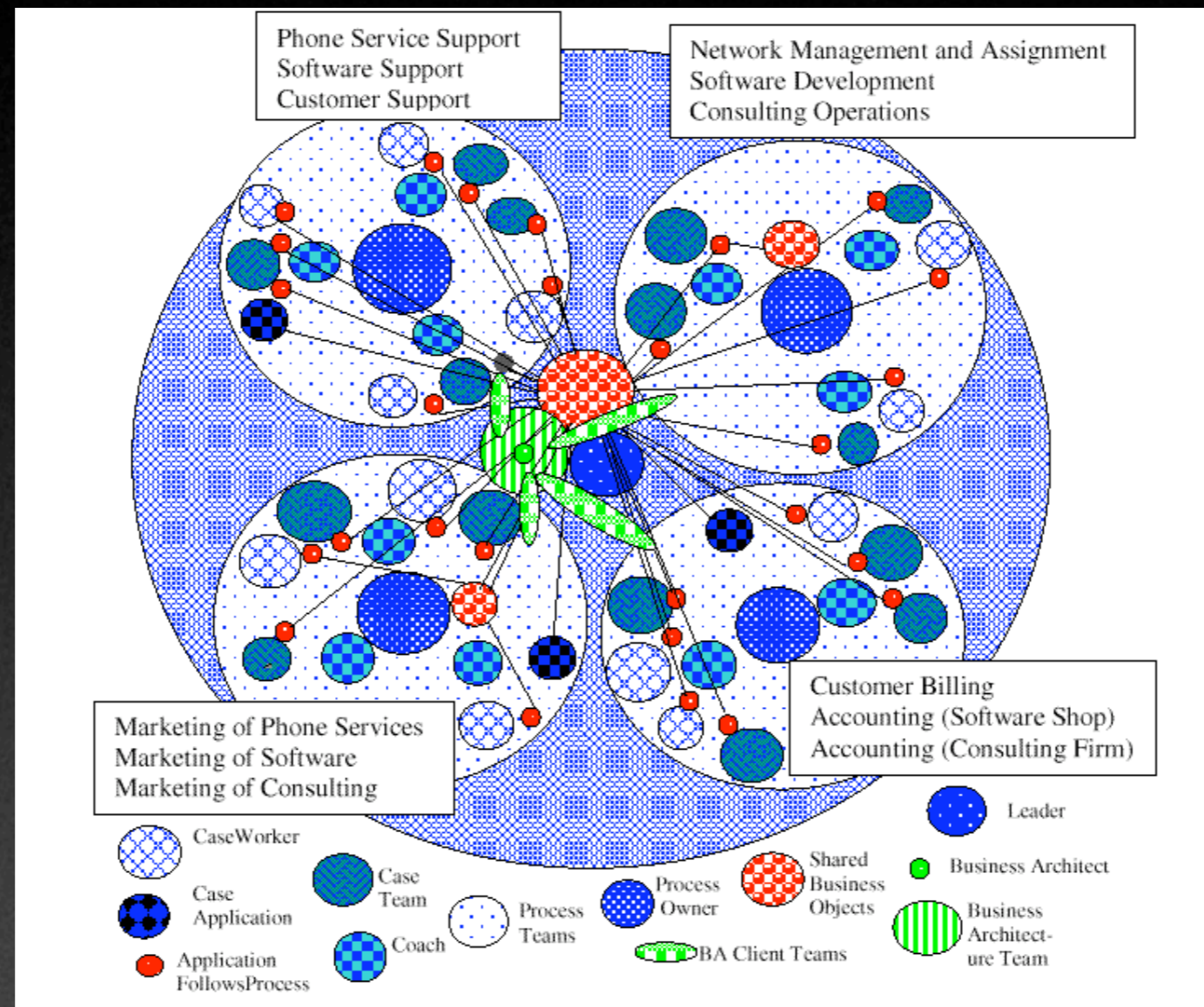
# Enterprise



# Agile



# Agile Enterprise



**Who are we?**



**Who are they?**



# Nature Network

- Social networking for scientists
- Capture relationship and publication data amongst academics and researchers
- Provide forum for online communication



**Team**

**28**

**People**



# What is Agile?

- Agile Manifesto (2001)
- Extreme Programming, Scrum, DSDM, Adaptive Software Development, Crystal, Feature Driven Development, Pragmatic Programming
- An alternative to “heavyweight” methods



# What is Agile?

1. Our highest priority is to **satisfy the customer** through early and continuous delivery of valuable software.
2. **Welcome changing requirements**, even late in development. Agile processes harness change for the customer's competitive advantage.
3. **Deliver working software** frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must **work together** daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and **trust them** to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is **face-to-face conversation**.
7. Working software is the primary **measure of progress**.
8. Agile processes promote **sustainable development**. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous **attention to technical excellence** and good design enhances agility.
10. **Simplicity**--the art of maximizing the amount of work not done--is essential.
11. The best architectures, requirements, and designs emerge from **self-organizing teams**.
12. At regular intervals, the team reflects on **how to become more effective**, then tunes and adjusts its behavior accordingly.



# What we actually do

- “lowercase agile”
- TDD + BDD / TATFT ^ XP
- Elements of “Getting Real”



# What we actually do

- Basically lightweight, goal-driven development
- Find good ideas, and embrace them



# What is Enterprise?

**“An undertaking,  
especially one of  
some scope,  
complication, and  
risk.”**



# What is Enterprise?

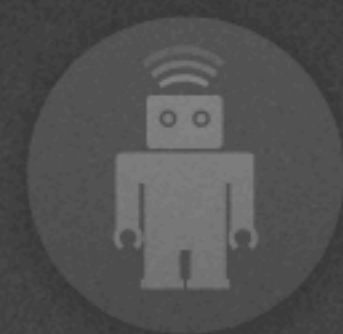
- A bunch of meetings
- Weird spec documents
- Lots of layers of management
- Constant re-organization



# What is Enterprise?

- Bureaucracy and internal politics
- People who “just” want a job





# Our goals

- Stay in business
- Make a profit
- Grow revenue
- Grow company



# Our real goals

- Have fun
- Work on interesting projects
- Learn constantly
- Find new challenges
- Productivity as a way of life



# Their goals

- Stay in business
- Grow revenue
- Preserve profitability
- Retain customers



# Their real goals

- Pay their mortgages
- Get kids to school on time
- Deliver quality software to customers
- Manage risk
- Go to meetings?





# Goals

- Are these cultures at odds with each other?
- What is the conflict?
- Where is the agreement?
- When do you put your foot down?
- When do you “just get it done”?



# Examples

- This is theoretically a case study so I should talk about the actual project
- Activity, Address, Affiliation, AffiliationMembership, Announcement, ApiKey, Authorship, Blog, Comment, Contribution, Discussable, Email, Event, Forum, Friendship, Group, Hub, Invitation, Job, Location, Message, Moderation, Photo, Post, Publication, PublicationInvitation, Reply, ReplyActivity, Speaker, State, Topic, User, and plenty more...



# Crazy routes

- Seemingly arbitrary preference
- No leverage despite difficulty
- Future headaches (500+ line routes file)
- Eventually (2 years later) came around



# Tickets & SCM

- JIRA
  - Allegedly an issue tracker
  - Really a way of life
- Perforce
  - Centralized SCM
  - Control vs Trust



# “Editor’s Blog”

- News vs blogs
- The speciallest of cases
- Implemented many times over
- Article, Blog, Post, News, Essay



# Key Steak Holders



# Ad Sales

- Make sure there's space to put ads
- Deploys should coincide with new sponsorships
- Sell then build



# Web publishing

- Publish more content
- Using “old” assumptions
- User generated content is weird



# Editors

- Write blogs
- Reply in forums
- Communicate with readers



# IT Group

- Keep this crazy ruby stuff away from our Java app servers.



# Developer Group

- Keep these crazy ruby developers away from our Java code.



# Product team

- Build a new brand for site
- Build a new community
- Satisfy business goals
- Preserve integrity of old brand



# Executive

- Expand brand without destroying it
- Make € and \$ from sponsors



# System Architects

- Integrate new tech and new product into 10+ years of architectural decisions



# QA Team

- Poor guidance
- Lots of power



# Analytics

- Track traffic
- Build “reports”



# Design team

- Preserve existing web brand
- Maintain markup consistency



# Finger Pointing



# Developers

- Obsessed with test coverage, not product quality
- Obsessed with semantic/valid markup, not site usability
- Obsessed with code elegance and refactoring, not improving site performance



# But, wait!

- Code quality, maintainability, elegance, semantics - these are all important!
- Well yeah, so is everything else all those other people are doing.



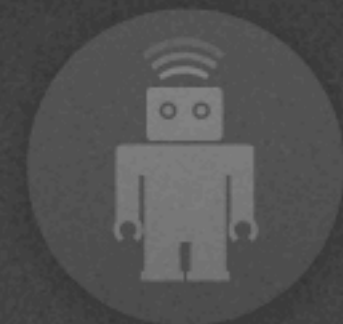
# Roles

- Developers are not “just coders”
- Developers must have constant access to product team
- Product team not allowed to interrupt developers



# Results

- 40K LOC, 1:2.4 Code:Test ratio.
- Deployed every other week for over a year, never rolled back, very few issues in live code.
- Viewed internally as instance of a high-quality “example” project for how things can go well and should be done.
- Led to more work for us (I think).



# Conclusion

- Lightweight process, productivity as a style, don't compromise, hire excellent people.
- Enterprise is about avoiding risk, managing change.



