





Your Game is Live, Now What?

Lessons Learned in the Online World



***Jane Fraser, QA Director,
Electronic Arts***





Pogo

- The site draws on average 16.5 million unique visitors per month (worldwide) who play for more than 56 minutes a day.
- Club Pogo has over 1.7 million paying subscribers.
- Club Pogo players spend on average 102 minutes a day on the site, 12 hours per week.
- The stickiest site on the web.





Intro

- Pogo is a casual game site from EA
- Both a free & subscription service with a community focus
- QA Director (15 + years in QA, 11+ in mgmt)
- 4 Offices – California, China, India, Canada
- 6 Managers & 45 Full Time & 44 Contractors (US, Canada, India, China, Vietnam)





About Pogo

- On average 16.5 million unique visitors/mth
- A free & subscription service
 - 100+ free games with a community focus
- Club Pogo
 - 40+ exclusive games
 - has over 1.7 million paying subscribers.
 - players spend on average 102 min/day.
 - 12 hours per week.
 - over 62% of Pogo players are female & 35+
 - the stickiest site on the web.





Roadmap

- Production Problems & Patching
- Release Planning
- Branching
- Release Plans
- Day to Day Issues
- Communication
- Post Mortems
- Monitoring





Overview

- Managing a successful site and trying to keep it bug-free is a never-ending process...
- Every release brings new challenges.
- Even external releases bring challenges (new IE versions, java, etc.)

The following is how Pogo is currently handling the inevitable issues that arise.





Take Aways

- Go Slow to Go Fast
- Plan for what you can
- Learn from your mistakes





24/7 support

- **Structuring your team for 24/7 support**
- Rotations, don't burn out your people.
- Training & Support.
- Planning (Plan for the worst).

If you don't juggle resources correctly,
the stress of managing a live site
will take a toll on your team.





The Sky is Falling

- I think I've found a problem, what do I do?
- **Think!**
- What is the user impact? Volume of users impacted? How long has it existed?
- How urgent is it? Business Impact?
- Assign a tester to try and reproduce.
- Assign an engineer to diagnose & fix.
- Go slow to go fast.





Patching Process

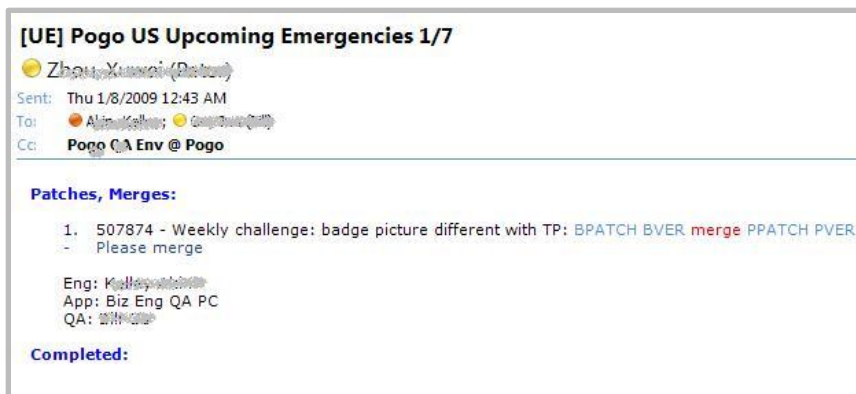
- We want to patch, now what.
- Determine the urgency.
(immediately, tonight, next week, never).
- How to fix it. (& user impact of fix)
- What is the risk of fixing.
- Approval process (QA, Eng, Production Control, Business Owner, Code Reviewer)
- Flow: Code Review, Test Environments, Production, Verification





Live Fixes

- Dealing with making fixes on production.
- Even the most benign fixes have risks.
- Approvals and Reviews templates.
- Automate manual tasks.
- Communicate



Patch Request Template
<p>Patch Request Template</p> <p>This is the template for sending a request for approval and scheduling of a new em</p> <p>The subject line for your email will be "Patch Request <environment> <bug number></p> <p>Copy this into your email and replace the ??? with the correct approver. Cut the "F</p> <p>(and NOT to production) then you should remove all Approval lines except the QA</p> <hr/> <p>Bug number and title of bug: ??? Environments affected: ??? What services will need to be restarted?: ??? Engineer who will fix the problem: ??? Engineer who will review fix: ??? Is the solution already in hand?: ??? If not, estimate when it will be available: ??? Approval (QA Manager): PENDING or APPROVED Approval (Engineering Manager): PENDING or APPROVED Approval (Change Reviewer): PENDING or APPROVED Approval (Production Control): PENDING or APPROVED Approval (Business Owner): PENDING or APPROVED User impact of problem: ??? Percentage or volume of users impacted: ??? Business impact of problem: ??? How long has the problem existed: ???</p>





Processing Patches

- Decide to fix
- Determine how to fix (code change)
- Review the fix
- Determine how to test the fix
- Review the test plan
- Test it out in the test environment
- Roll it out to production
- Test again
- Now you can focus on why it happened.





Release Planning

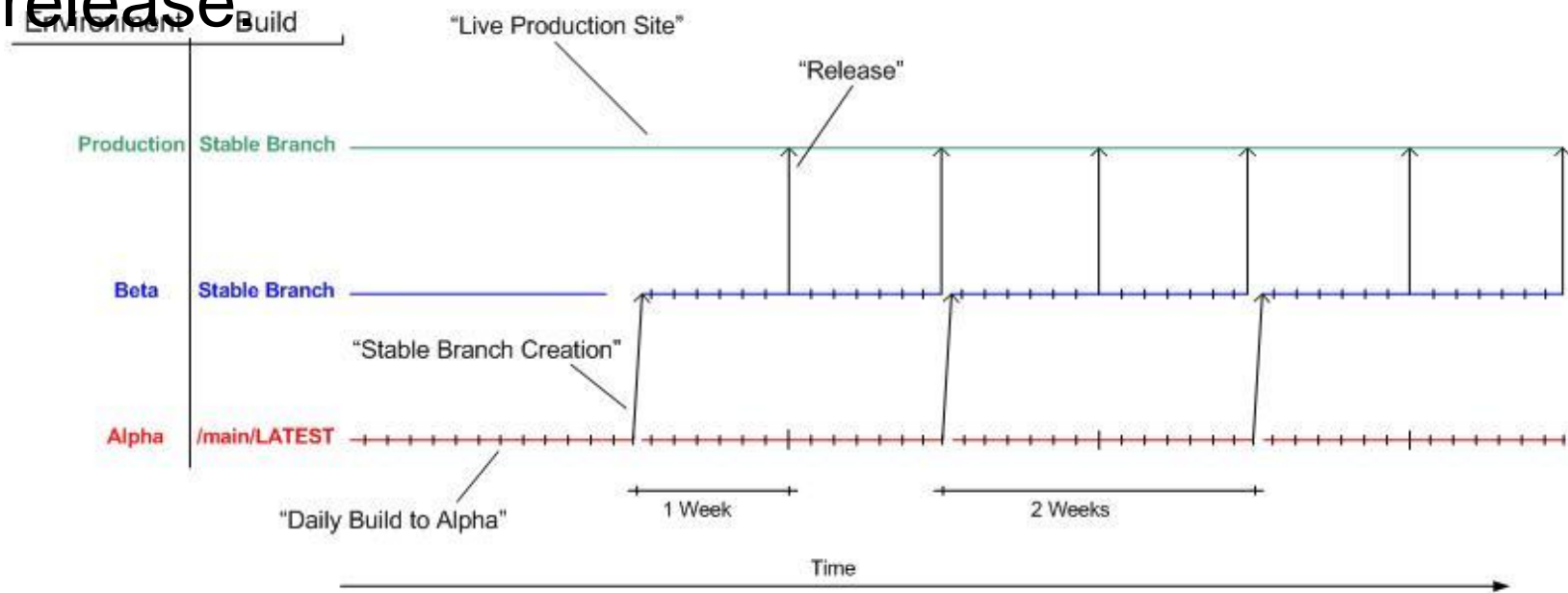
- Pogo uses a Bus analogy to get products on production.
- With a multi-branch system and several early test environments.
- All code moves through test > staging > live.
- Once your code is on main, its on the bus for a short ride to production.





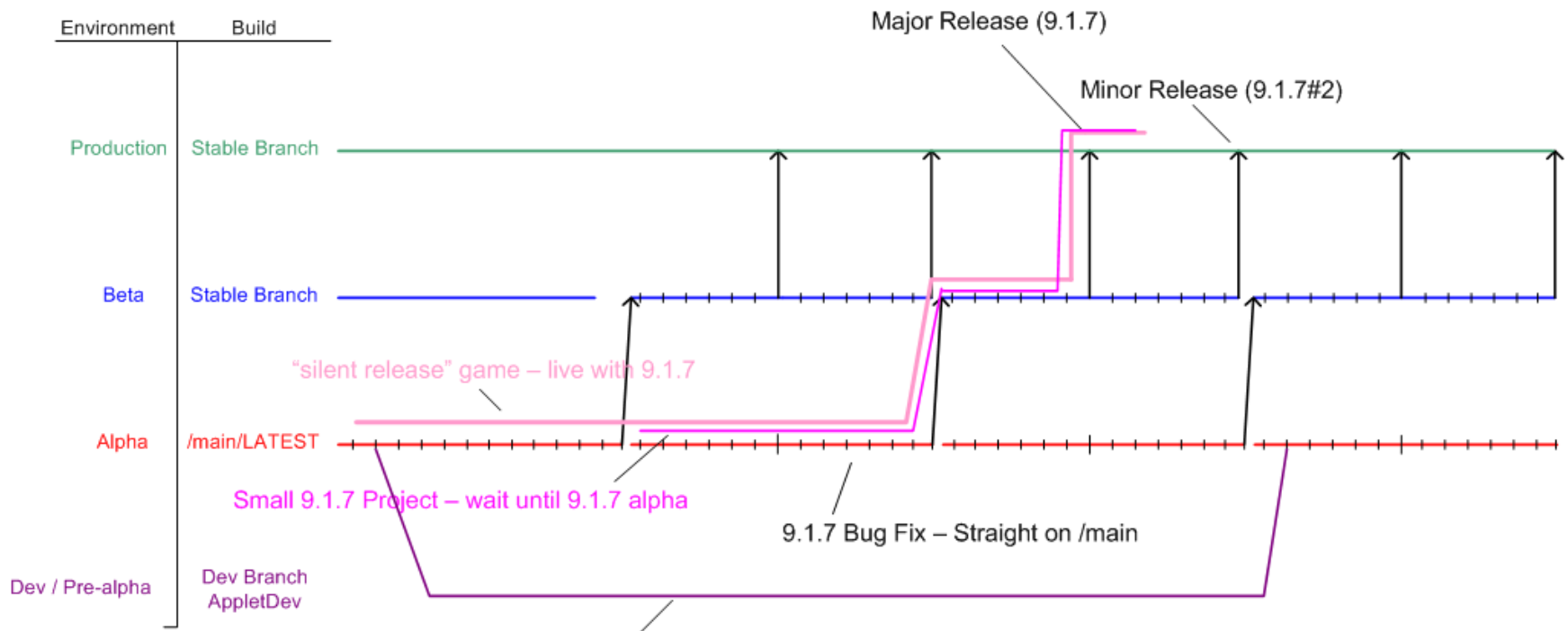
Release Branching

- We use a lot of branches to give us more testing time on projects.
- Decreases the risk of a project affecting a release





Branches





Release Meetings

- Our releases are between 5 & 20 projects.
- Detailed list of changes & who is in charge of each change & dependencies and timing.
- QA's confidence in releasing each project.
- Full work plan is created for every release.
- Who is on duty and who is on call.





The Release

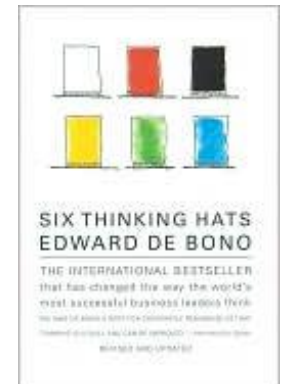
- Both a call bridge and chat room are started.
- The work plan is run through.
- QA runs automated and manual regression tests. New projects have there own tests which get run.
- Any issues found at this point use the production troubleshooting procedures.
- Release Status Report is sent out at the end of the verifications.





Mitigating risky releases.

- Learn from the past
- War rooms
- Check lists
- Great time to use your 6 thinking hats.





War Rooms



Norad War Room

The Ultimate War Room



The War Room - the operational crisis centre - is established at the gateway to disaster, at the foot of volcanoes or the World Trade Center towers. When disaster strikes, it bolts from cover with its sirens blaring, just like the fire brigade. It organizes, calms, directs and comforts. It attempts heroically to deal with the unmanageable, the chaos...



Pogo's War Room



Don't forget the **Food**,
Remember your "army"
travels on its stomach

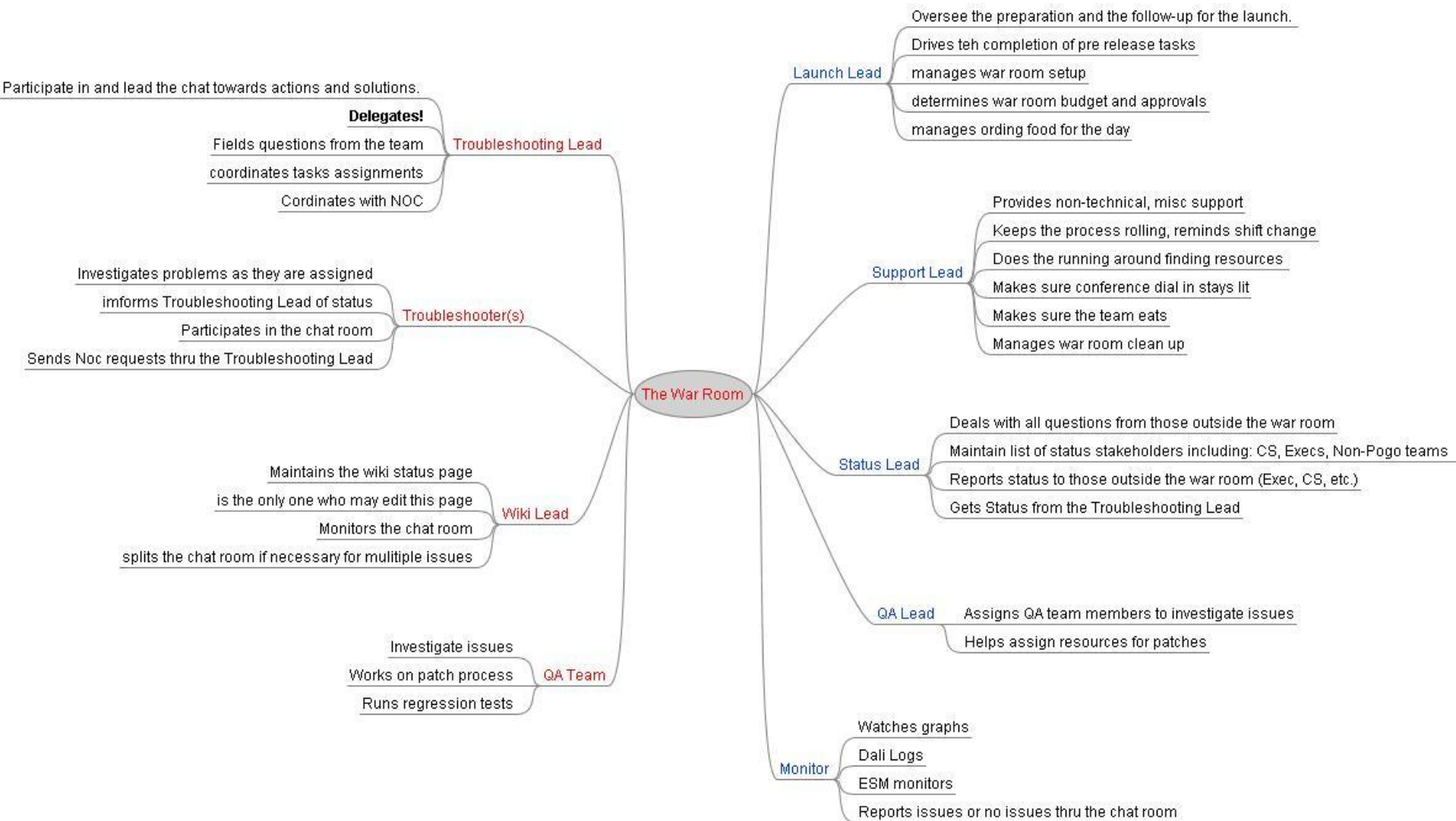




War Room

- For our riskier release we schedule 2 days
- Shifts are 6 hours in length
(2am – 8am, 8am – 2pm, 2pm – 8pm, 8pm - 2am)
- Prevent burnout
- Learn the ropes by shadowing
- Both a chat room and conference calls are used all day







For the Day to Day

- Prod troubleshooting rotation
- Still with the war room tasks, but on a slim basis
- 2 engineers & 2 QA.
- Weekly team, Lead & Secondary for each.
- Secondary is there just in case Lead is awol.
- The rest of the team focuses on the future





Prod Troubleshooting Team

- Create the schedule well in advance to ensure availability.
- Juggle people around as needed.
- Wiki Status Page for each week.
- Email alias to streamline issues.





Communication & tools.

- Run books
- Wiki's – Current Status
- Chat Rooms, IM
- Conference Calls and Rooms
- War Rooms
- Schedules and roles





Conventions & Contacts

- Style guides – for bugs, patches, names

<SO-UNIX_mohit>	step 5 completed
<SO-DBA_Amar>	DB and listeners on 400a are
	please proceed
<So-Unix-Manish>	ok can i proceed for rolling re:
<SO-DBA_Amar>	yes, as db and listeners are up
<So-Unix-Manish>	ok

- Make sure to have all your contact info readily available

Web browser screenshot showing the Pogo Wiki page titled "Contacts - Pogo". The page content includes:

Contacts - Pogo
Refer to the main [Contacts](#) page for all of Pogo.

Contacts - Non-Pogo
Note: the list of contacts below may not be current. Verify prior to project launch. If th find all the contacts needed in the main [Contacts](#) page.

Contacts - Online Commerce Publishing (EA Link, Commerce)

Name	Phone	Role
Lily Weiss	408-308-8875	Business Lead
Shirley Chen	408-613-9394	Tech Lead

Contacts - Online Publishing (CS, CRM, VCMT)

Name	Phone	Role
Lily Weiss	408-308-8875	Business Lead
James Whitcomb	434-982-1657	Tech Lead

Contacts - Online Publishing (JIVE, EREG/OREG)

Name	Phone	Role
------	-------	------





Post Mortems

- Learn from your successes and mistakes.
- Review: How could we have done better?
- How could we have caught it sooner?
- Sometimes there isn't a cost effective solution.





Monitoring

- Monitoring from the QA point of view.
- Bug reports
- Check lists
- Logs
- Release Notes
- Usage Graphs

Production Alerts From The Last Hour

Top 10:	Prod Pogo	Prod Besl
All Over 200:	Prod Pogo	Prod Besl
All New:	Prod Pogo	Prod Besl
New Over 200:	Prod Pogo	Prod Besl
All:	Prod Pogo	Prod Besl
Raw XML:	Prod Pogo	Prod Besl
Immediate Alerts:	Prod Pogo	Prod Besl

Production Alerts From The Last 10 Minutes

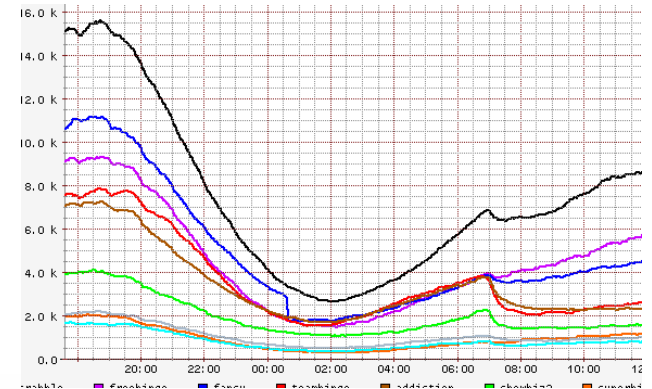
Top 10:	Prod Pogo	Prod Besl
All Over 75:	Prod Pogo	Prod Besl





Understand Your Customer

- Why you need to understand your customer.
- You need to know natural trends (Dinner breaks).
- American Idol finales takes its toll on our users.
- When we start challenges on games, everyone flocks to those games.





Summary

- Create processes and procedures to help you maintain your site.
- Learn from your past and others.
- Plan for the worst.
- Enjoy your successes.
- Continuously review.
- Always look for better ways.





Questions?





Thank You

- If you have any questions please feel free to contact me.
- Jfraser@ea.com
- <http://www.linkedin.com/in/qaguru>

