

# Role of Retrospectives in Success of Agile Projects

Bapiraju Nandury  
Curam Software



# Objective



- Not “just-another-boring-theory-session”!
- Information YOU can use
- Interactive





- **Introduction to Retrospectives**
  - What?
  - Why?
- **Case Study**
  - Why did we feel a need for this practice?
  - What challenges did we overcome?
- **How to run effective Retros?**
  - Ground rules
  - Tips and Tricks

**cūRAM**<sup>®</sup>  
SOFTWARE

# Introduction to Retrospectives



# Introduction to Retrospectives



- “Project Management is like driving a car”
  - Continuous (and instantaneous) correction based on feedback!



Hey, Baps! Time to prove yourself!! Great project for you - no clear requirements nor allocated resources. Get the job done by next week, dude!



# Introduction to Retrospectives



- What is a Retrospective (rèt´re-spèk-tîv)?

“A ritual held at the end of a project to learn from the experience and to plan changes for the next effort”

<http://www.retrospectives.com>



- **What is a Retrospective?**

**“Retrospective is an event held at the end of a period with measurable deliverable(s) to review the history, learn from experiences and plan for the future. It’s intended to discover, share and pass along the learning experience”**

**Bapiraju**



# Introduction to Retrospectives



- **Purpose**

- Very old idea, not a new one
- Learn from past experiences (success/failure)
- “Is your organization good at acquiring and using its wisdom in creating software?” - Norman Kerth
- “Making lemons from lemonade” - Norman Kerth



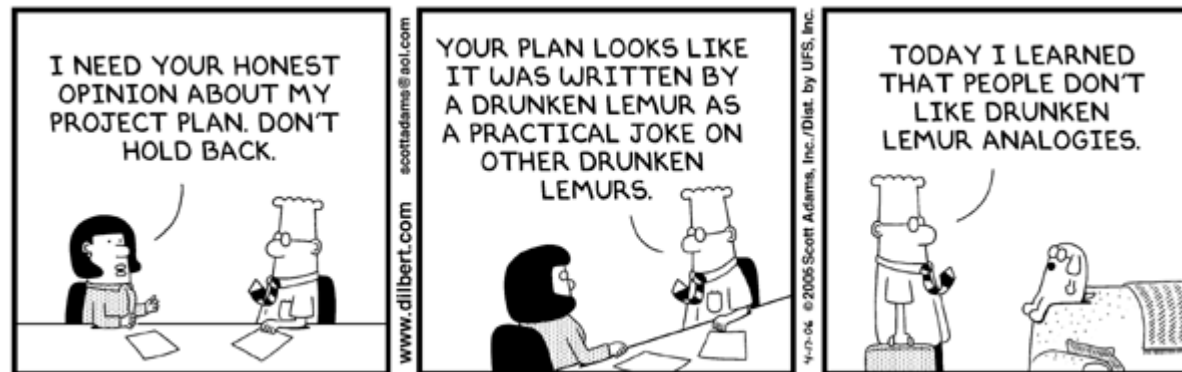
# Introduction to Retrospectives



- Prime directive

“Regardless of what we discover, we understand and truly believe that everyone did the best job they could, given what they knew at the time, their skills and abilities, the resources available, and the situation at hand”

Norman Kerth



© Scott Adams, Inc./Dist. by UFS, Inc.

# Introduction to Retrospectives



- **Ground Rules**

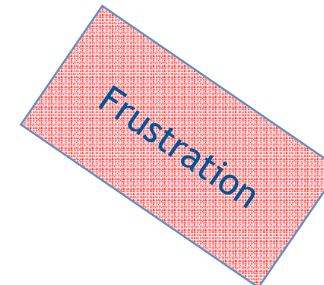
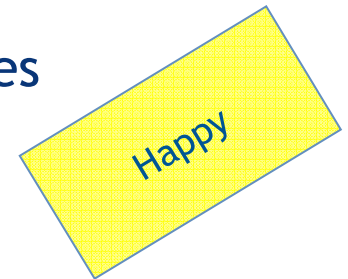
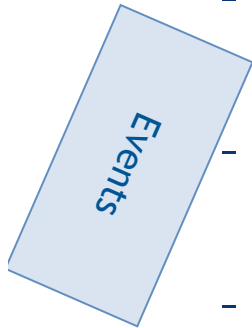
- We will try not to interrupt each other
- We will accept everyone's opinion without judgment
- We will talk from our own perspective, not from anyone else's
- We will turn off our mobile phones
- Participation is optional

# Introduction to Retrospectives



- What happens in a project retrospective?

- Facilitator kick-starts the session, explains objectives
- Manager gives concise account of events occurred
- Color-coded cards filled in by the team
- Answer 4 key questions and draw “Emotional Seismograph”
- Appreciation and wish-list



# Introduction to Retrospectives



- **The Four Key Questions\***
  - What did we do well, that if we don't discuss we might forget?
  - What did we learn?
  - What should we do differently next time?
  - What still puzzles us?

\*Source: <http://www.retrospectives.com>

cūRAM<sup>®</sup>  
SOFTWARE

# Retrospectives A Case Study



# Retrospectives - A Case Study



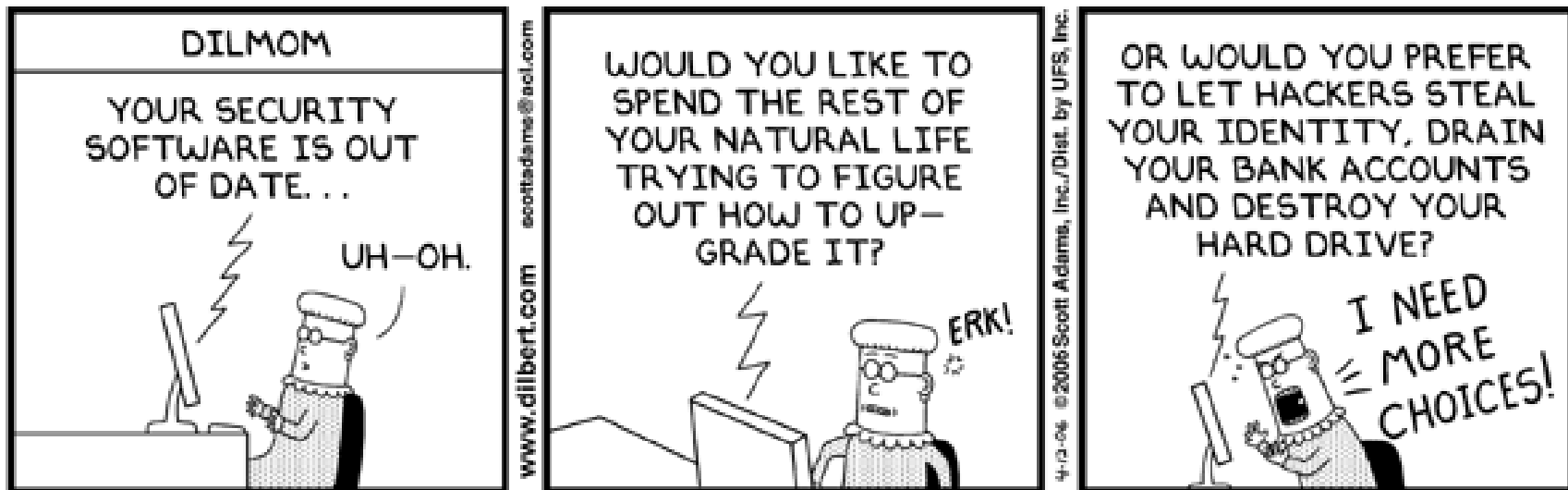
- **Project Context**

- Case study of a project in one of my earlier companies (2003-'04)
- Unhappy, forceful client
- Not “appealing” work for devs
- Brand new team - new hires
- Complex domain and lack of knowledge
- Confusion and frustration
- No “Different” from a typical offshore project! 😊

# Retrospectives - A Case Study



- We were desperately looking for ideas



© Scott Adams, Inc./Dist. by UFS, Inc.

- And we were told that there's a "choice"!



# Retrospectives - A Case Study



- Retrospectives practice was a break through for us
- Negative energy was converted into positive energy
- WYTIWYG - WhatYouThinkIsWhatYouGet 😊
- Table all the concerns and brainstorm for solutions
- Continually track progress (daily stand-ups)

# Challenges and their resolution



- **Configuration Issues**
- **Forced implementation**
- **Lack of confidence in the dev team**
- **Rotation of stakeholders**
- **Inadequate development processes**
- **Technical challenges**



- **Configuration Issues**

## Challenge

- Config issues to be resolved by client - as per contract - but, never done
- Classification of issues
- “Blame culture”

## Resolution

- Explicit communication
- Share data like hours spent fixing issues rather than actual dev
- Methodological approach to solve issues
- Proactive attitude

# Challenges and their resolution



- **Forced implementation**

## Challenge

- No technical expertise with client
- Suggestions that led us nowhere
- Forced implementation, that the team hated

## Resolution

- Found the need to increase level of confidence
- Over communication
- Technical solutions explained in plain English
- Reduce response time for questions

# Challenges and their resolution



- Lack of confidence

## Challenge

- Unpleasant history
- Client lost confidence in team
- Dev team was under pressure as even simple things had to be done with extreme care

## Resolution

- Communicate, communicate and **OVER COMMUNICATE**
- Transparency in process - daily status mails, wiki updates
- Share daily schedules with **ALL** the stakeholders
- Wiki

# Challenges and their resolution



- **Rotation of stakeholders**

## Challenge

- Stakeholders at client organization moved around without a proper transition plan
- Mostly BAs and QAs
- Communication gap

## Resolution

- Let client know about the problem
- Local resource - stepped up as “client”
- Explore ways to increase domain knowledge

# Challenges and their resolution



- Inadequate development process

## Challenge

- Complex domain
- Difficulty in getting started
- “No Documentation”!
- No crosspollination

## Resolution

- Follow agile - XP
- Rapid feedback
- Weekly Retrospectives  
- after closure of iteration
- QA pairing - between India and US for domain knowledge



# Challenges and their resolution



- **Technical Challenge**

## Challenge

- Rigid design
- Inflexible architecture
- QA servers not in sync
- Non-identical QA environments between dev and client, which made tests unreliable

## Resolution

- Pair programming to crack design issues
- Ask for help
- Phased solution for QA environment sync
- Test Automation

# How to run effective Retros?



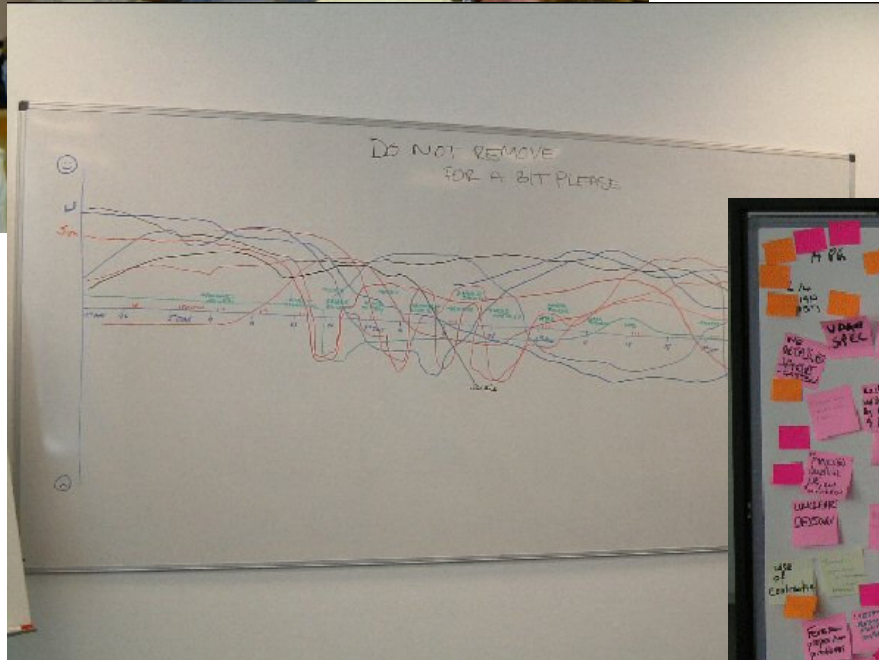
# Effective Retrospectives



The Big Picture

Source: <http://industrialxp.org>

Seismograph



Source: <http://www.agilexp.com>

Timeline



# Effective Retrospectives



- Prerequisites
  - Sell to the team
  - Ground rules
  - Agree on “Prime Directive”
  - Open-minded team
  - Experienced facilitator
  - Preparation

# Get the best out of Retros!



- **Tips and Tricks**

- Have courage to embrace change
- Collect data from project
- Create a “timeline” of events
- Use different colors for cards
- Try “Emotional Seismograph”, it was useful for us
- Do not encourage personal jokes or criticism

# Get the best out of Retros!



- **Tips and Tricks**

- Have an external facilitator
- Work aids - Charts, pens/pencils, color coded cards, tape etc.
- Post on walls
- Conduct retrospectives at the end of every iteration
- Keep the process simple and lightweight
- Build people, so that they can build projects (appreciate)



# Get the best out of Retros!



- **Tips and Tricks**

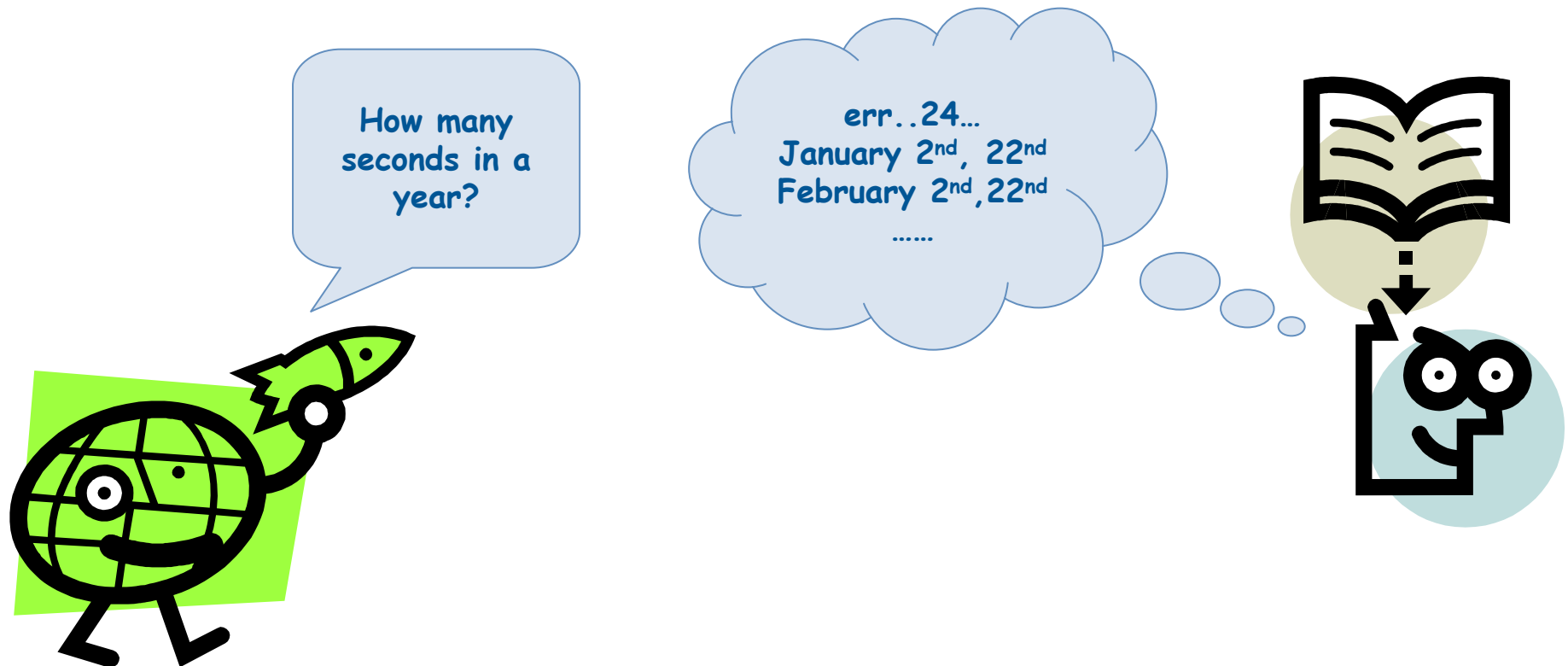
- Improve quality of project and also lives of people on it
- Encourage continuous learning
- Help to change perceptions
- Invite spectators from other projects
- Post results visibly - ideally where standup happens
- Do not lose focus



# Lessons Learnt



- A practice that we learnt, will never forget in life, will keep doing it for continuous improvement
- Think out-of-the-box and have fun



# References



- <http://www.retrospectives.com/>
- “Project Retrospectives: A Handbook for Team Reviews” by Norman L Kerth

# Thank You

Questions/Comments

[bnandury@curamsoftware.com](mailto:bnandury@curamsoftware.com)

<http://nandury.blogspot.com>

