Building Applications in Developing Nations

Dr. Paul Dorsey

Dulcian, Inc.

www.dulcian.com

Sept 28, 2010

The Task

- Budget and Finance System for the government of Ethiopia
  - 1000 sites, 5000 users
  - 20 languages
  - Replace a legacy system
    - SQL Server => Oracle
    - Complex => simple architecture
    - No change in user functionality

The Challenges

- Limited or no connectivity
  - Internet to the town but not to the buildings
  - “DonkeyNet”
  - Viruses everywhere
- Large area (2x Texas)
- Limited IT skills of government employees
- No senior IT skills available in country
- Dirty data in source system
- Cultural differences

You Can't Imagine

- Population 80M
- 85% in "agriculture industry"
- No health care, no dentists
- Elections are meaningless.
- Measures of GDP are meaningless.
  - Much of economy is barter based.
  - Top 20 poorest countries

- "Slum Dog Millionaire" is optimistic.

FAQ

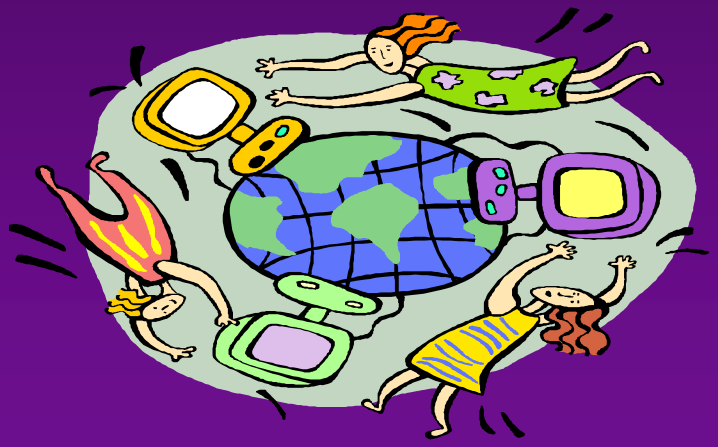
- Life in Ethiopia
  - Different but not terrible
  - Lots of servants
  - Lots of things to get used to
  - People are nicer than here.
  - Streets are safer.
- Life with the Bobert (my 4 year old)
  - “Daddy, what’s that smell?”
  - “Look, sheep costumes!”

Cool things

- Animals
  - Elephants, hippos, hyenas, crocs, camels
- Food
  - Fruits and veggies
  - Fresh meat (yes VERY FRESH)
- Weather
- The people

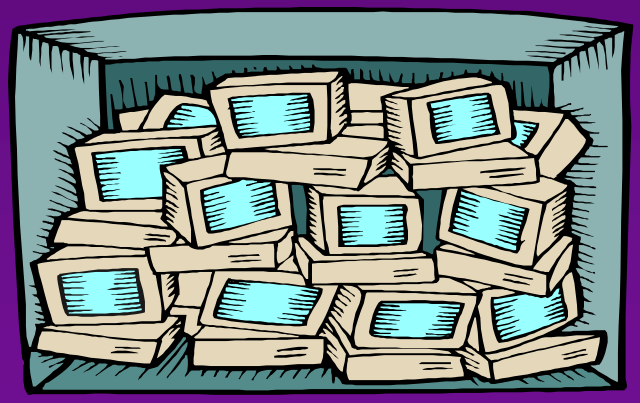
Is Ethiopia relevant?

- USA has bandwidth problems.
  - Upstate NY has poor bandwidth.
  - E-Business didn't work at Alaska sites.
- Java EE is a tough environment.
  - > 6 months to learn, 50% fail to learn
  - Evolving architecture
  - No consistency
  - High failure rate of projects
    - We are better at lying.



Assets Available

- Overpriced hardware
- Very smart developers
- A well-built V1 system as a model
- Free un/low-skilled labor
- Money (when they feel like it)

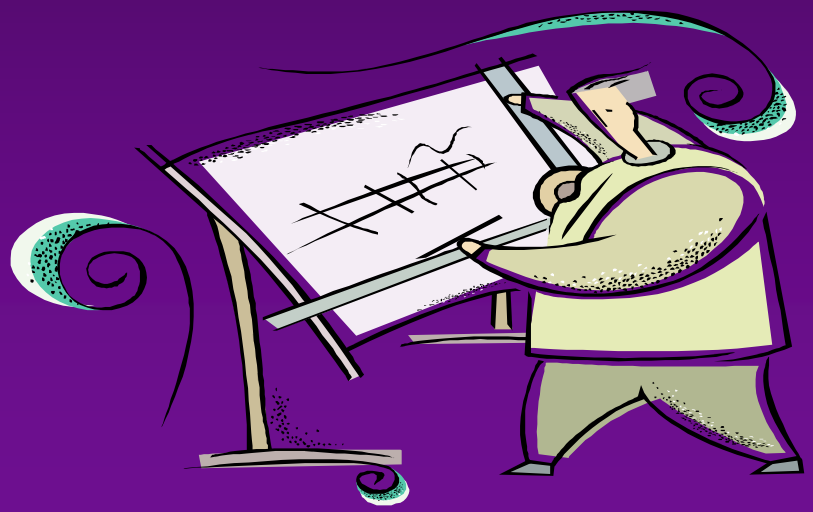


Why Not COTS?

- Too complex
- Too expensive
  - $25 million for 23 sites in Uganda
  - $300 million and counting in Pakistan
    - "I feel like I have invited the devil into my home and I can't get him to leave."
- Unreasonable page size
  - Web Center: "Some of our pages are less than 1 MB."
- Huge hardware requirements

New Web Architecture

- Low bandwidth
- Repository-based
- Thick database
- Event Action Framework
- SQL and PL/SQL only
  - Easy to learn
  - Productive

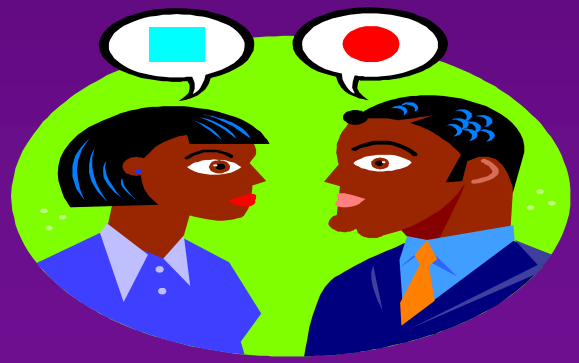


Project from Hell

- Ethiopian manager actively sabotaged project.
  - Divided organization
  - Destroyed client relationship
  - Tried to steal project
- No trust from client
  - No cooperation from users
  - Resistance from testers
- Government internal issues
  - Culture of finger pointing

Client Culture

- Foreigners respected but not trusted
  - By both client and my staff
- Internal client culture includes finger pointing
  - Functional managers compete
- Managers vary in experience
- Client IT dept is understaffed
- No project champion
  - Who is in charge?



Project History

- Year 1
  - Build new architecture
  - Rebuild system
- Year 2
  - Testing (with massive delays)
  - New Module
  - Data migration and cleanup

Status

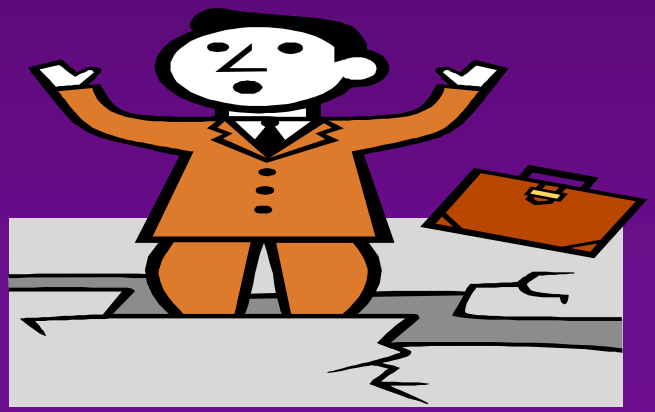
- Architecture
  - Great success
- Rebuild
  - Had to stay consistent with existing system
- Testing
  - Finally good
  - Best tested system ever
- Deployment
  - Client refuses to plan.

Successes

- Great success with system architecture
- Well built V2
- Well tested system
- Rebuilt trust of client - mostly

Failures

- Took too long
- Project scope was too big
- Didn’t deal with internal attack in time
- Lost trust of client

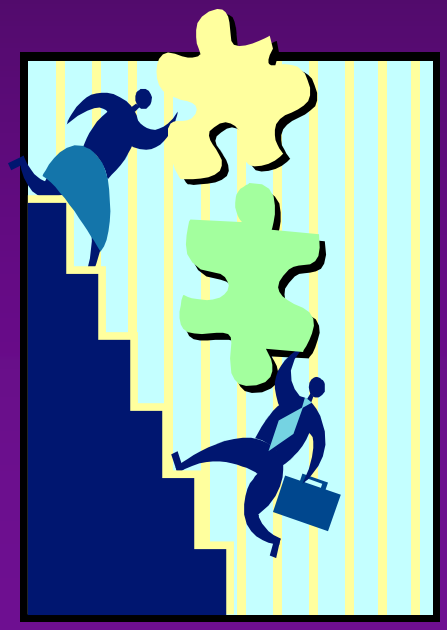


Lessons Learned

- Smaller scope
- Trust no one
- Challenges are…
  - Technical
  - Organizational
- New architecture was great.

Next Steps

- Find/Create local Technical Manager
- Find/Create local Project Manager
- Continue to support architecture design
- Build on customer relationship



What did I learn?

- If you can build it there… you can build it anywhere
- Everything is clearer.
- Required:
  - Good architecture
  - Good management
  - A smart staff
  - A good working environment