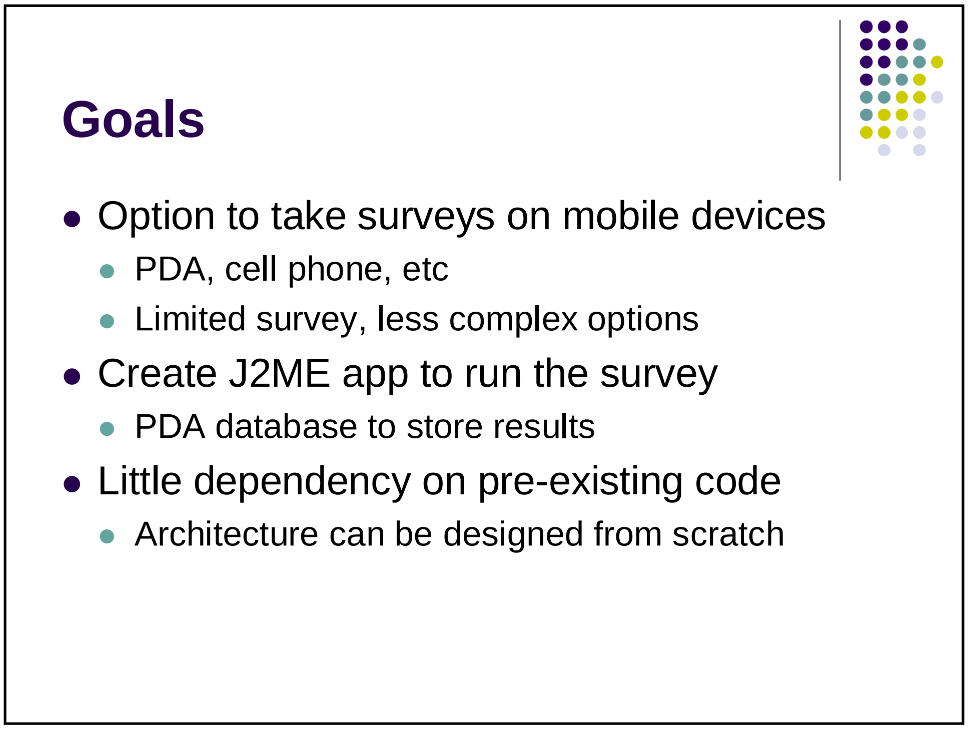
J2ME Survey

Group IV

Salahuddin Choudhary, Yanka Li, Colin Rothwell, Key Shin

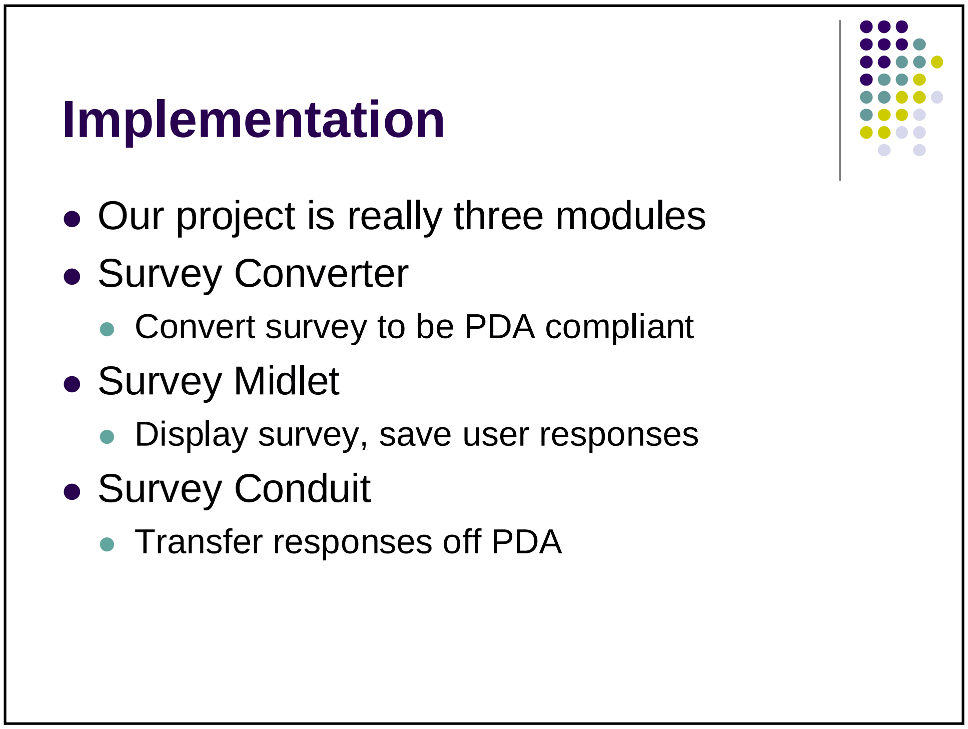
Client

- Aaron Powers
  - Research Associate, HCI Institute
- Web Survey Toolbox
  - Create surveys for academic experiments
  - Java-based survey builder tool
  - Generates JSP-driven HTML surveys
  - Many questions types, advanced options



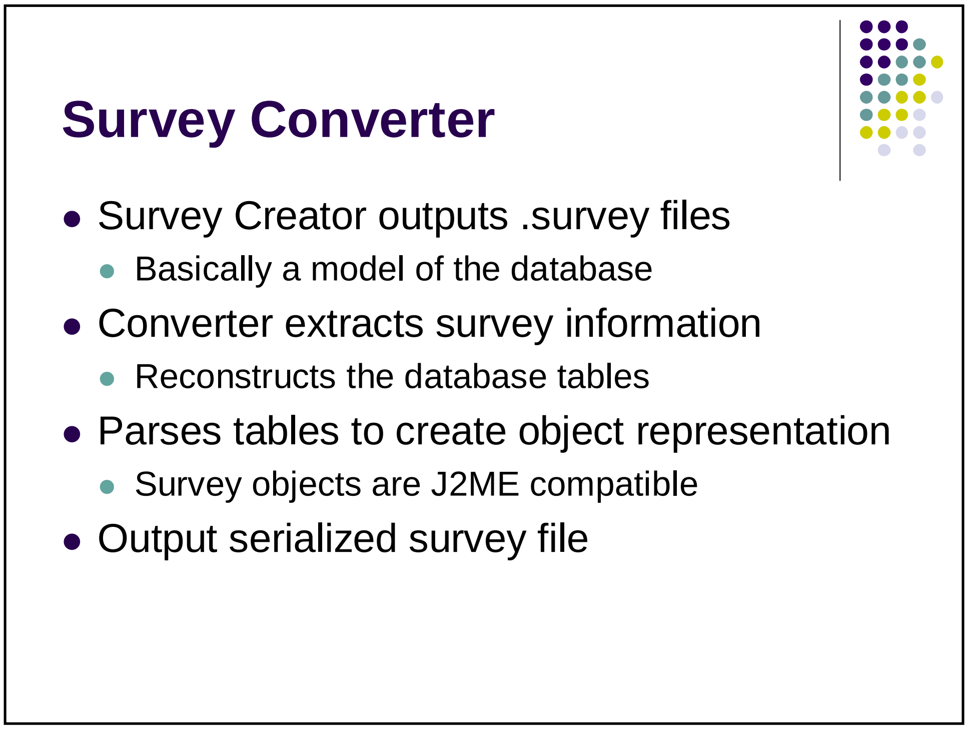
Goals

- Option to take surveys on mobile devices
  - PDA, cell phone, etc
  - Limited survey, less complex options
- Create J2ME app to run the survey
  - PDA database to store results
- Little dependency on pre-existing code
  - Architecture can be designed from scratch



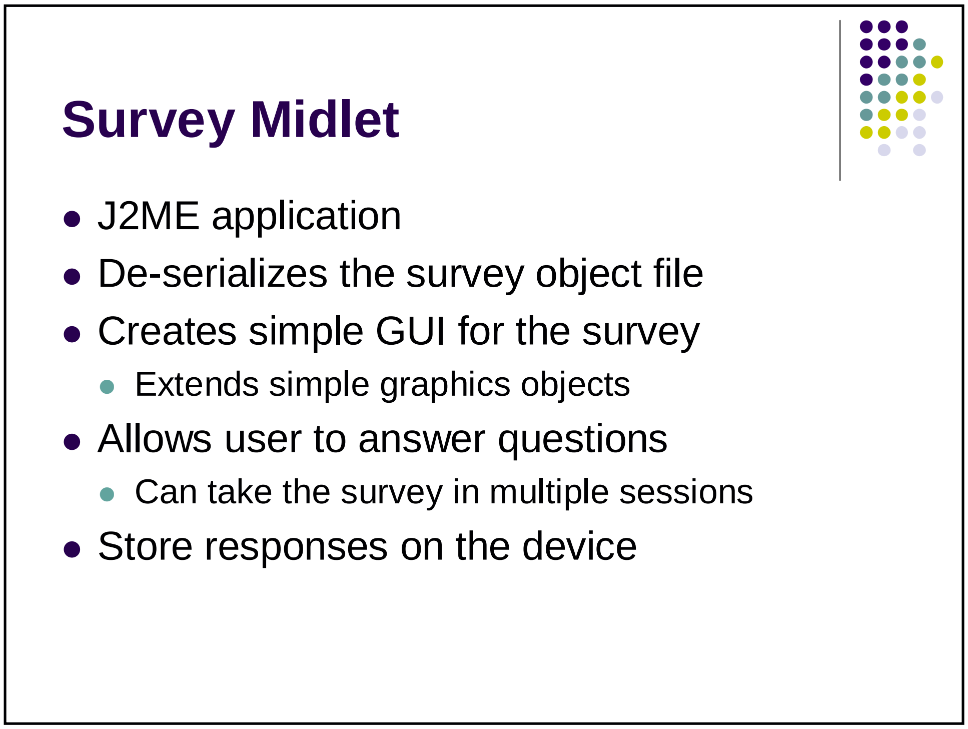
Implementation

- Our project is really three modules
- Survey Converter
  - Convert survey to be PDA compliant
- Survey Midlet
  - Display survey, save user responses
- Survey Conduit
  - Transfer responses off PDA



Survey Converter

- Survey Creator outputs .survey files
  - Basically a model of the database
- Converter extracts survey information
  - Reconstructs the database tables
- Parses tables to create object representation
  - Survey objects are J2ME compatible
- Output serialized survey file



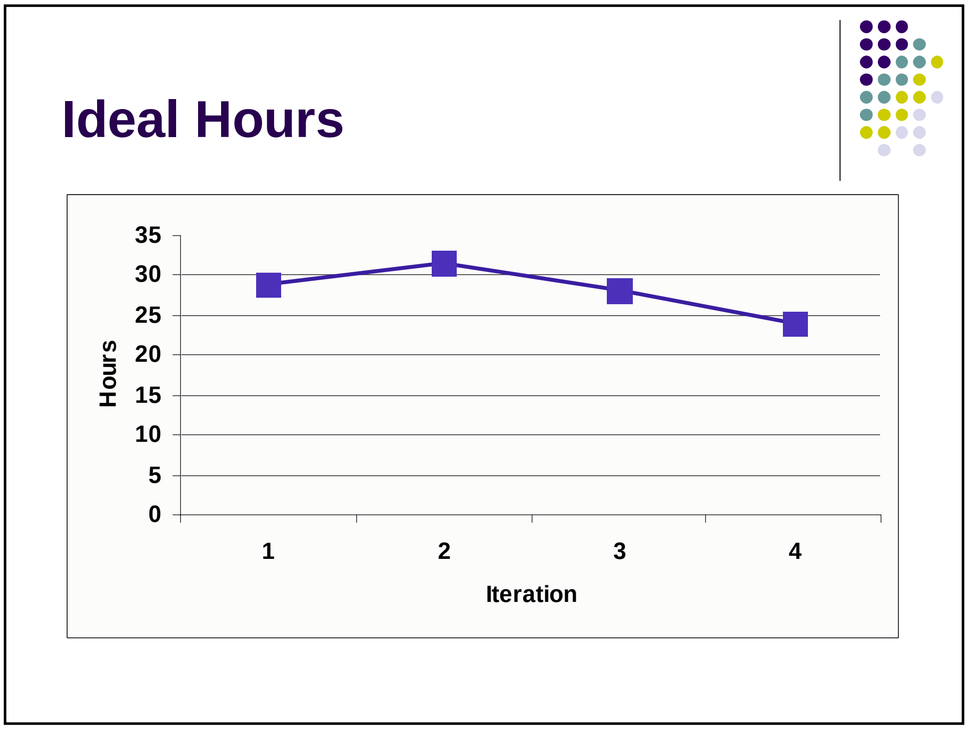
Survey Midlet

- J2ME application
- De-serializes the survey object file
- Creates simple GUI for the survey
  - Extends simple graphics objects
- Allows user to answer questions
  - Can take the survey in multiple sessions
- Store responses on the device

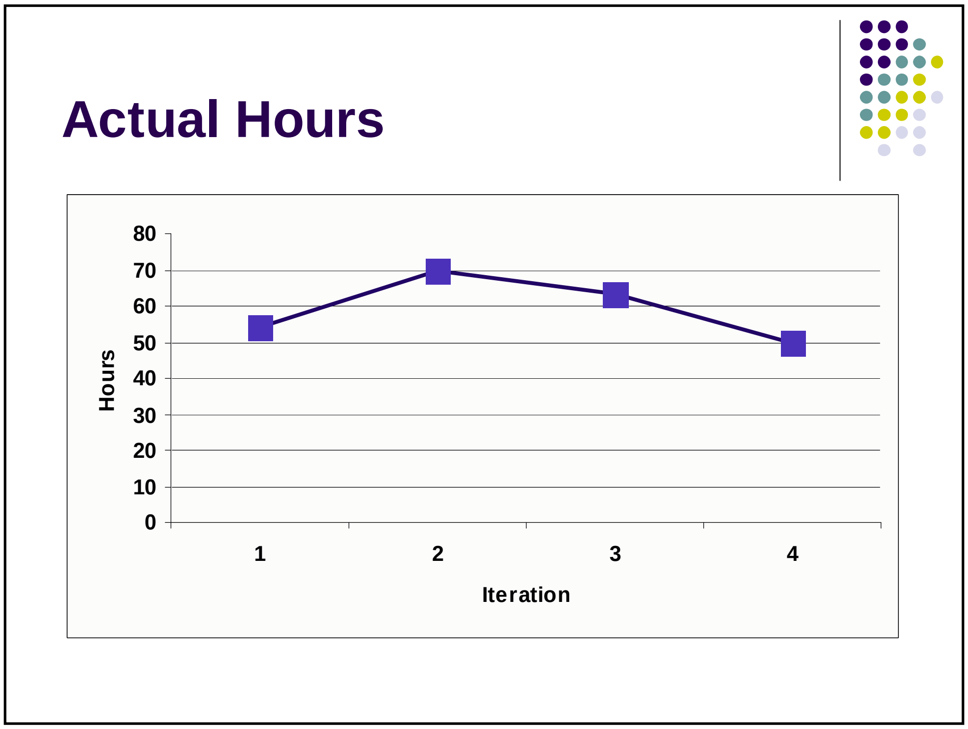
Survey Conduit

- Program that runs during Palm HotSync
  - Specify configuration file on install
- Two different purposes:
  - Fetch answers from Palm databases and store as .csv files
  - Delete previous databases

Demonstration

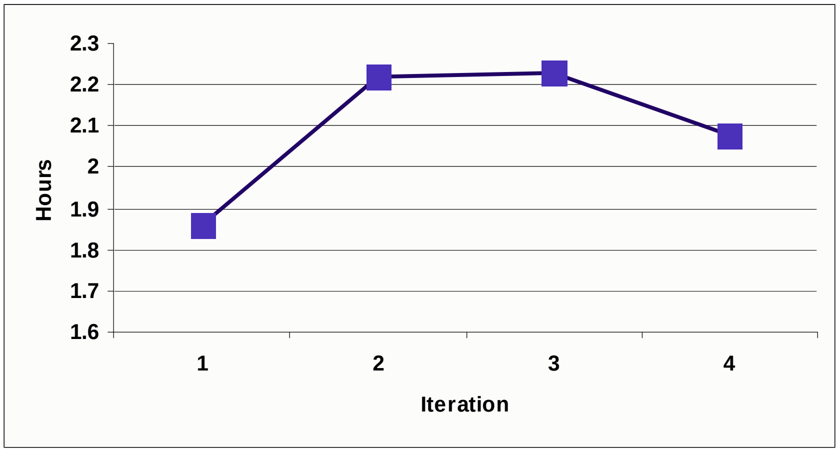


Ideal Hours



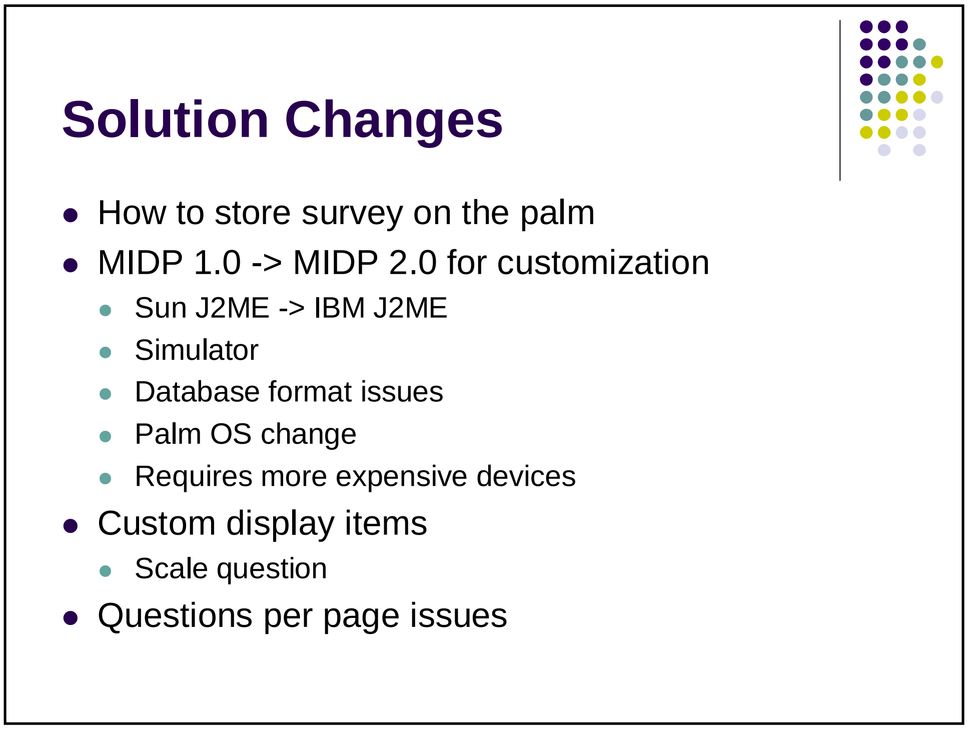
Actual Hours

Load Factor



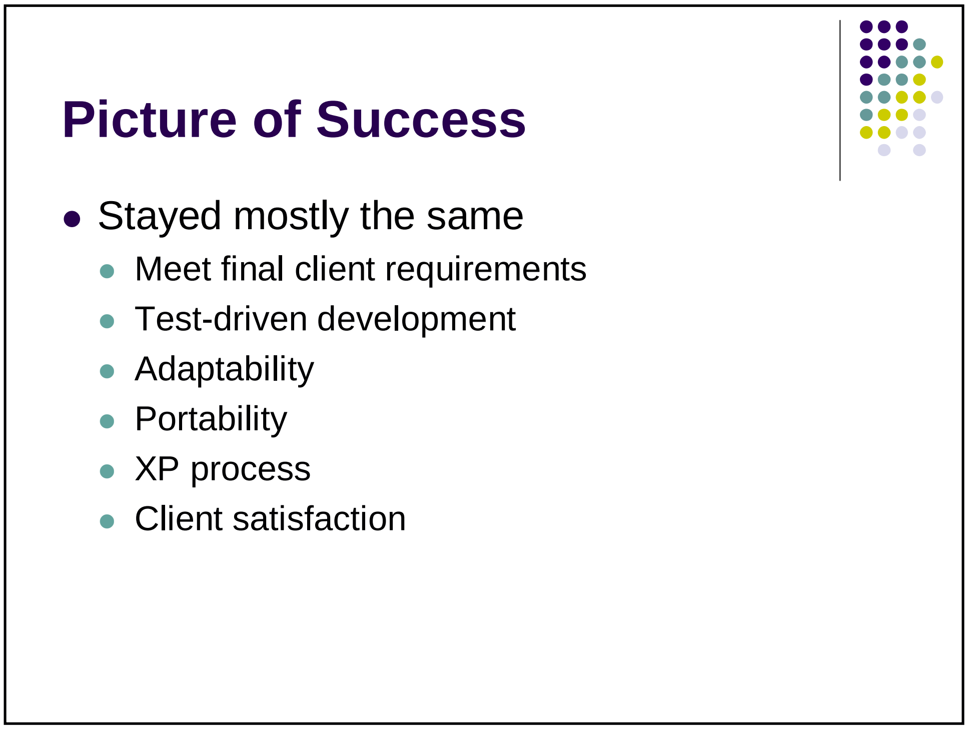
Requirement Changes

- JDBC support issue
- Decoupling of parsing and converting
- No support for abstract question
- CSV files -> Survey files
- Ant build file -> Java
  - Exporting directly to PRC
- Create a conduit installer



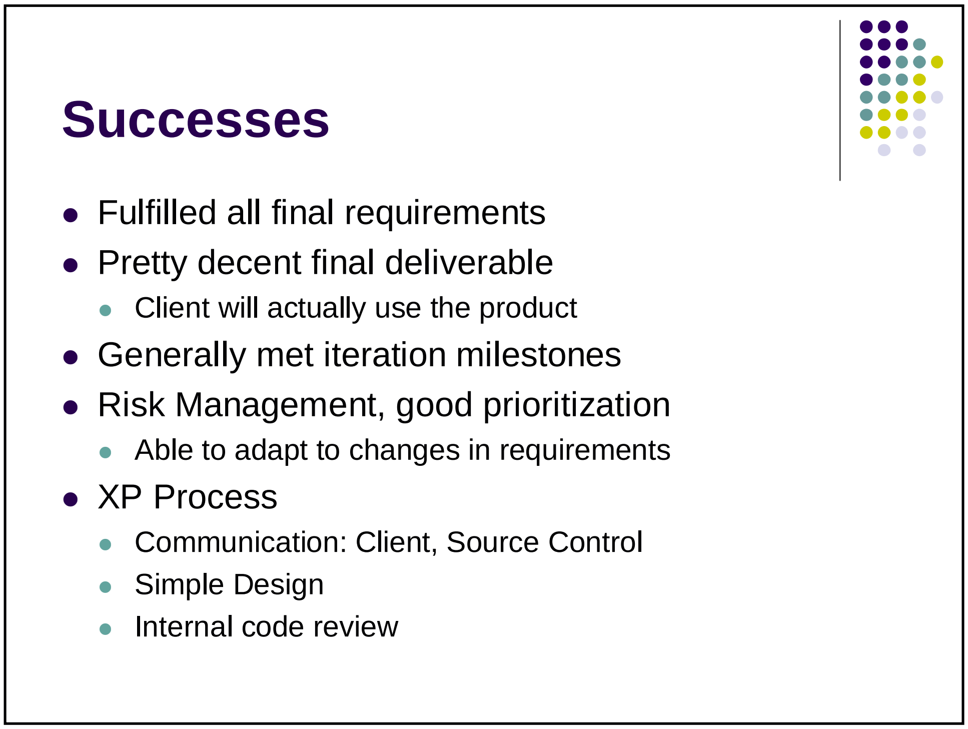
Solution Changes

- How to store survey on the palm
- MIDP 1.0 -> MIDP 2.0 for customization
  - Sun J2ME -> IBM J2ME
  - Simulator
  - Database format issues
  - Palm OS change
  - Requires more expensive devices
- Custom display items
  - Scale question
- Questions per page issues



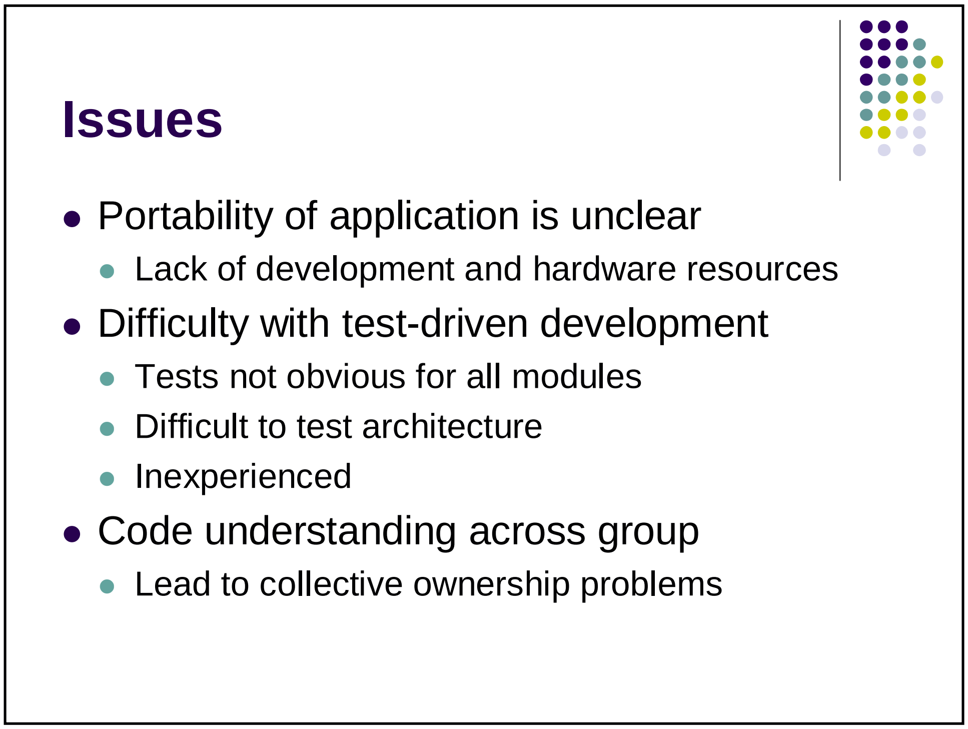
Picture of Success

- Stayed mostly the same
  - Meet final client requirements
  - Test-driven development
  - Adaptability
  - Portability
  - XP process
  - Client satisfaction



Successes

- Fulfilled all final requirements
- Pretty decent final deliverable
  - Client will actually use the product
- Generally met iteration milestones
- Risk Management, good prioritization
  - Able to adapt to changes in requirements
- XP Process
  - Communication: Client, Source Control
  - Simple Design
  - Internal code review



Issues

- Portability of application is unclear
  - Lack of development and hardware resources
- Difficulty with test-driven development
  - Tests not obvious for all modules
  - Difficult to test architecture
  - Inexperienced
- Code understanding across group
  - Lead to collective ownership problems

XP: Positive

- Many principles useful
  - Contributed directly to success of project
- Exploration phase
  - User stories, gathering requirements
  - Risk analysis
- Whole team
  - Customer needs/specifications changed
  - Need vs. reality
- Simplicity
  - Build working simple solution and expand

XP: Negative

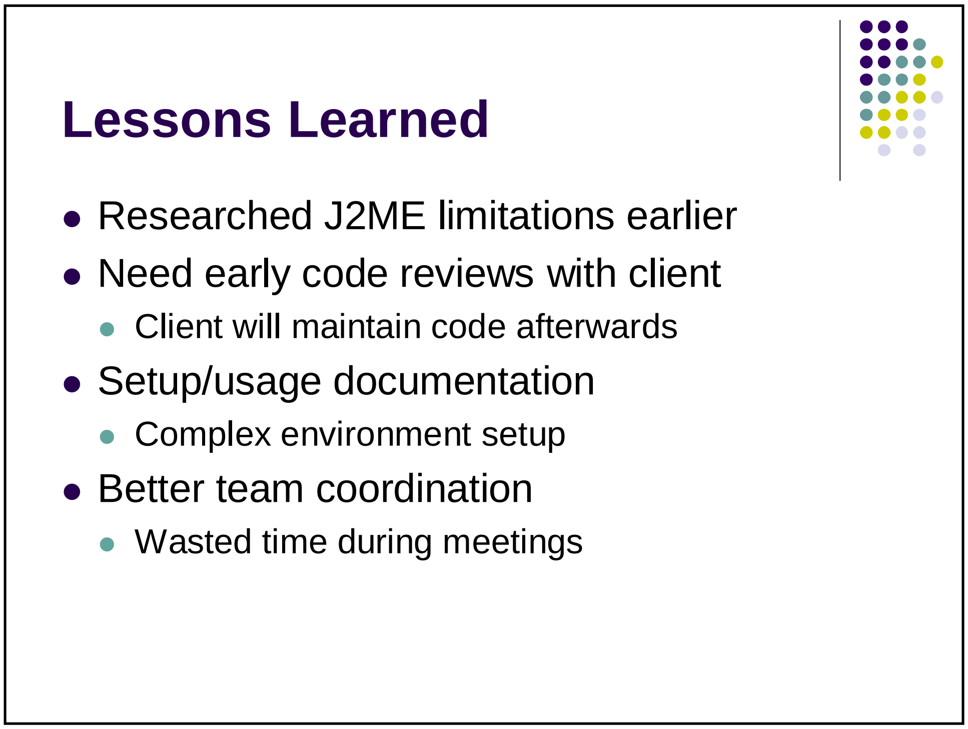
- Principles sometimes a hindrance
- Collective ownership
  - Leads to confusion between members
  - Better communication about changes
- Documentation
  - Sometimes code is not immediately apparent
  - Client will be the maintainer
- Pair Programming

Pair Programming

- Not conducive to academic environment
  - Scheduling issues
  - Geographic pairs
- Friends aren’t always great partners
- Disparity in skill can be frustrating
- Alternative: Code Review
  - When we couldn’t schedule pair meetings

Other Engineering Topics

- Wideband Delphi
  - Best meeting ever
  - Varying results of success
- Prototyping
  - Throwaway GUI application
- Design
  - Modularity
  - Information Hiding



Lessons Learned

- Researched J2ME limitations earlier
- Need early code reviews with client
  - Client will maintain code afterwards
- Setup/usage documentation
  - Complex environment setup
- Better team coordination
  - Wasted time during meetings