The FAO Open Source Cadastre & Registration Software Project

Experts Open Source Workshop
Rome November 2010
## Advantages of Computerisation:

- land, rights and ownership information should be more accurate and accessible
- related office processes can be stream-lined and made more efficient
- land records can be stored more securely, without deterioration (climate & animals) and with backups which allow disaster recovery

Which results in improved security of tenure
How can computerised systems be made more sustainable?

- Minimize external software support dependency (especially beyond period of development assistance)
- Reduce ongoing software licence & support costs
- Maximize user confidence & competence
- Ensure reliability of system backup processes
  - Select hardware that is appropriate for the operating environment
1. Why Computerise?

2. Open Source Software (FLOSS)

3. FAO FLOSS Cadastre & Registration Project

How can computerised systems facilitate good governance?

- Improve transparency in land records and related processes
- Reduce service delivery delays & easier monitoring of service standards
- Reduce discretionary decision making through more automated checks / controls
- Sophisticated controls reducing the chance of accidental or malicious changes to land records
- All changes to key land records are recorded and record changes can be easily audited
Why use open source software?

1. Why Computerise?

- Gives more freedom to utilise other developer’s software and you do not have to “re-invent” software where it already exists

2. Open Source Software (FLOSS)

- Gives more freedom to devise an approach to software development that suits your circumstances

3. FAO FLOSS Cadastre & Registration Project

- Potential cost savings if sustainability issues are appropriately addressed
Advantages of Open Source (FLOSS):

- Why Computerise?
  - Ongoing software licence costs
- Open Source Software (FLOSS)
  - Vendor “Lock-in”
  - Based on “open” data formats and international standards
  - Open Source software can co-exist with some proprietary software used in cadastre & registration
  - Well designed Open Source software can address sustainability issues
  - There is a range of sophisticated and mature Open Source DBMS & GIS products that could be utilised within FLOSS cadastre & registration projects.
Support to the Development & Piloting of Open Source Cadastre & Registration Software Project

- unded by the Government of Finland
- year project
- stage project

1. Planning, documenting requirements & design
2. Software development of initial generic software
3. Customisation & piloting in 3 countries (Samoa, Nepal & Ghana). Inclusion of other pilot countries is possible if additional funding can be found
- Establishment of active developer & user
FAO FLOSS Cadastre & Registration Project
Gantt Chart
## Current Project Activities

| 1. Why Computerise? |  
|--------------------|---|
| 2. Open Source Software (FLOSS) |  
| 3. FAO FLOSS Cadastre & Registration Project |  

- Why Computerise?
- Definition of Requirements
- Software Architecture
- Register as Open Source Project
- Identify Open Source Resources
- Identify Software Standards & Protocols
- Software Quality Plan
- Establish Central Software Development Team
- Plan Open Source Training in Pilot Countries
1. Why Computerise?

2. Open Source Software (FLOSS)

3. FAO FLOSS Cadastre & Registration Project

Project Roles (Central):

- Project Steering Committee
- Project Sponsors
- Internal) Peer Reviewers
- External Quality Review
- Project Manager / Coordinator
- Academic Partner
Project Roles (Ghana, Nepal & Samoa):

- Local Coordinator
- Lead Coordinator
- Steering Committee
- Working Group
- Local Software Development Team
- International Software Developer (Short – term)
1. Why Computerise?

2. Open Source Software (FLOSS)

3. FAO FLOSS Cadastre & Registration Project

- FAO Project Document
- ilot Country Feasibility Study Reports
- Inception Report
- Monthly Progress Reports
- Generic Business Requirements (Part 1)
- Generic Software Detailed Requirements (Part 2)