soundsoftware.ac.uk

Sustainable Software for Audio and Music Research

Mark Plumbley, Chris Cannam, Luis Figueira

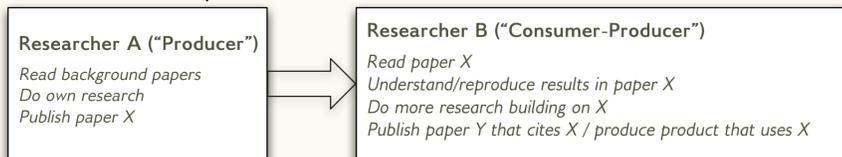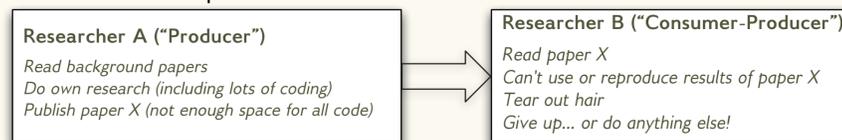Centre for Digital Music, Queen Mary University of London

About the SoundSoftware.ac.uk project

Provide a **service**:
▷ to the audio and music research community
▷ to support development and reuse of software and data
▷ to enable high quality reproducible research

Ideal Research Pipeline

Researcher A ("Producer")	→	**Researcher B ("Consumer-Producer")**
Read background papers		*Read paper X*
Do own research		*Understand/reproduce results in paper X*
Publish paper X		*Do more research building on X*
		Publish paper Y that cites X / produce product that uses X

Real Research Pipeline

Researcher A ("Producer")	→	**Researcher B ("Consumer-Producer")**
Read background papers		*Read paper X*
Do own research (including lots of coding)		*Can't use or reproduce results of paper X*
Publish paper X (not enough space for all code)		*Tear out hair*
		Give up... or do anything else!

Strategy:
▷ Employ software developers to make existing research software robust and usable
▷ Train researchers to write more reliable research code
▷ Help academics and research project managers to build software development into research projects
▷ Curate data and software to help future researchers find what they need

Identify and address barriers to publication and reuse of code

How we are addressing barriers to code publication and reuse

Barrier: Lack of confidence with code

Our approach:
▷ Workshops
▷ Software Carpentry for Audio & Music Research (2010)
▷ Get researchers to work on their code together
▷ A wide range of workshop formats planned for 2012
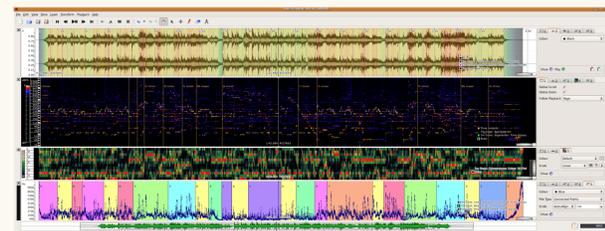
Barrier: Lack of facilities and tools

Our approach:
▷ Provide a code hosting site
▷ Support version control (using Mercurial)
▷ Support public & private projects for sharing with other researchers
▷ Provide EasyMercurial, a simple to teach and learn user interface for version control
▷ Develop learning material: videos, tutorials

Barrier: Platform incompatibilities

Our approach:
▷ Use a plugin architecture for code distribution
▷ Produce standard code modules and rework into more than one target format
▷ Take advantage of existing popular application ecosystems

Barrier: Lack of incentive for publication

Our approach:
▷ Associate publication references with software
▷ Make it easier for readers of publications to find code...
▷ ... and for anyone seeking the code to know what to cite
▷ Thus increasing citation impact