GAMMA
Continuous Evolution of Software After Deployment

Alessandro Orso, Donglin Liang,
Mary Jean Harrold, and Richard Lipton

College of Computing
Georgia Institute of Technology

Supported by NSF CCR-9988294, CCR-0096321,
EIA-0196145, Boeing Aerospace Corp,
State of Georgia Yamacraw Mission

Motivation

Developing reliable software is difficult

- Software's intrinsic and increasing complexity
- Limits in the application of testing and analysis techniques

Situation has worsened

- Increased demand due to widespread use
- Increasingly complex environments

Need ways to

- Analyze software after deployment
- Find and fix problems in the user's environment
- Collect accurate usage data

Outline

- Overview of GAMMA
- Experience with GAMMA
- Related work
- Conclusion

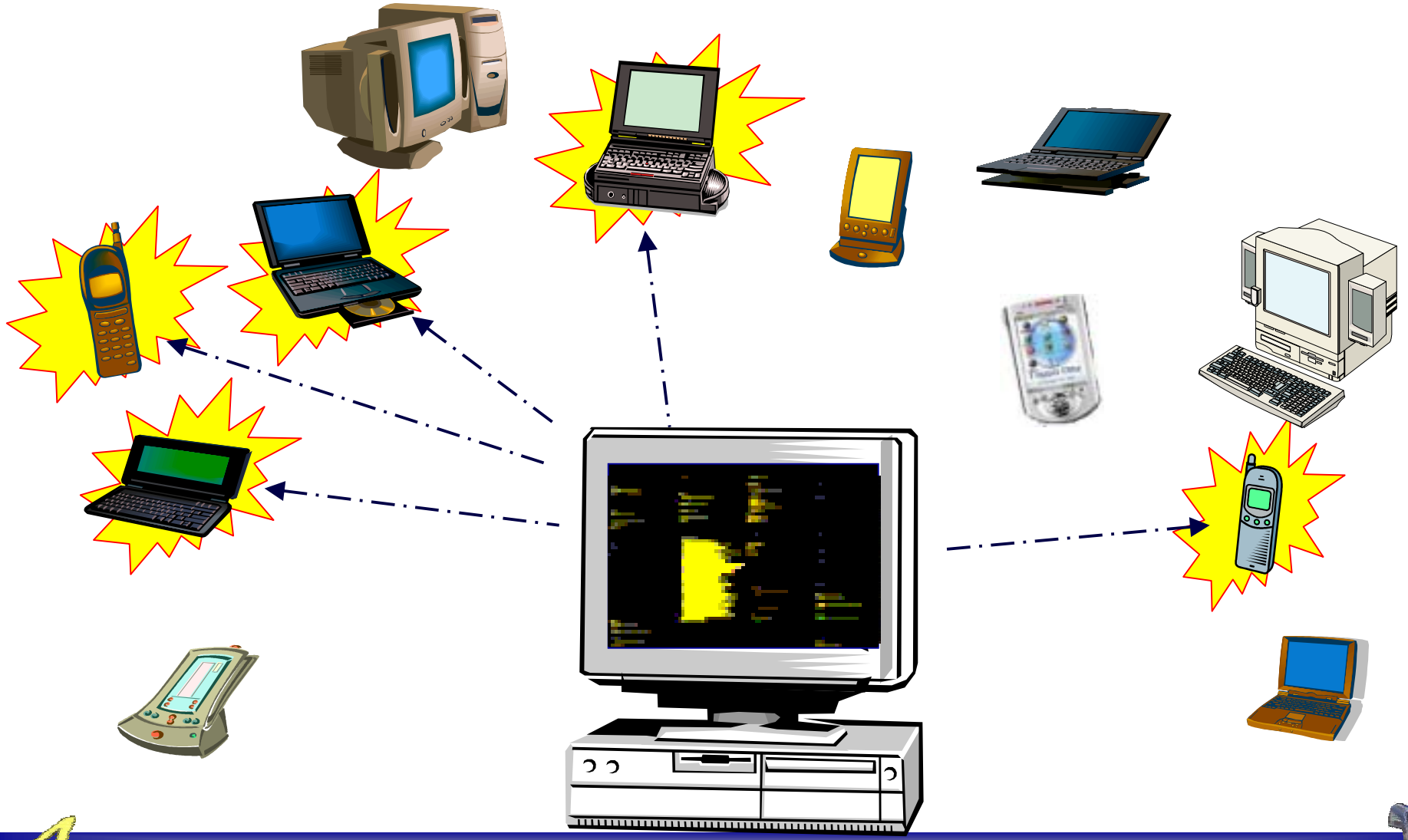
GAMMA

A technology that lets developers

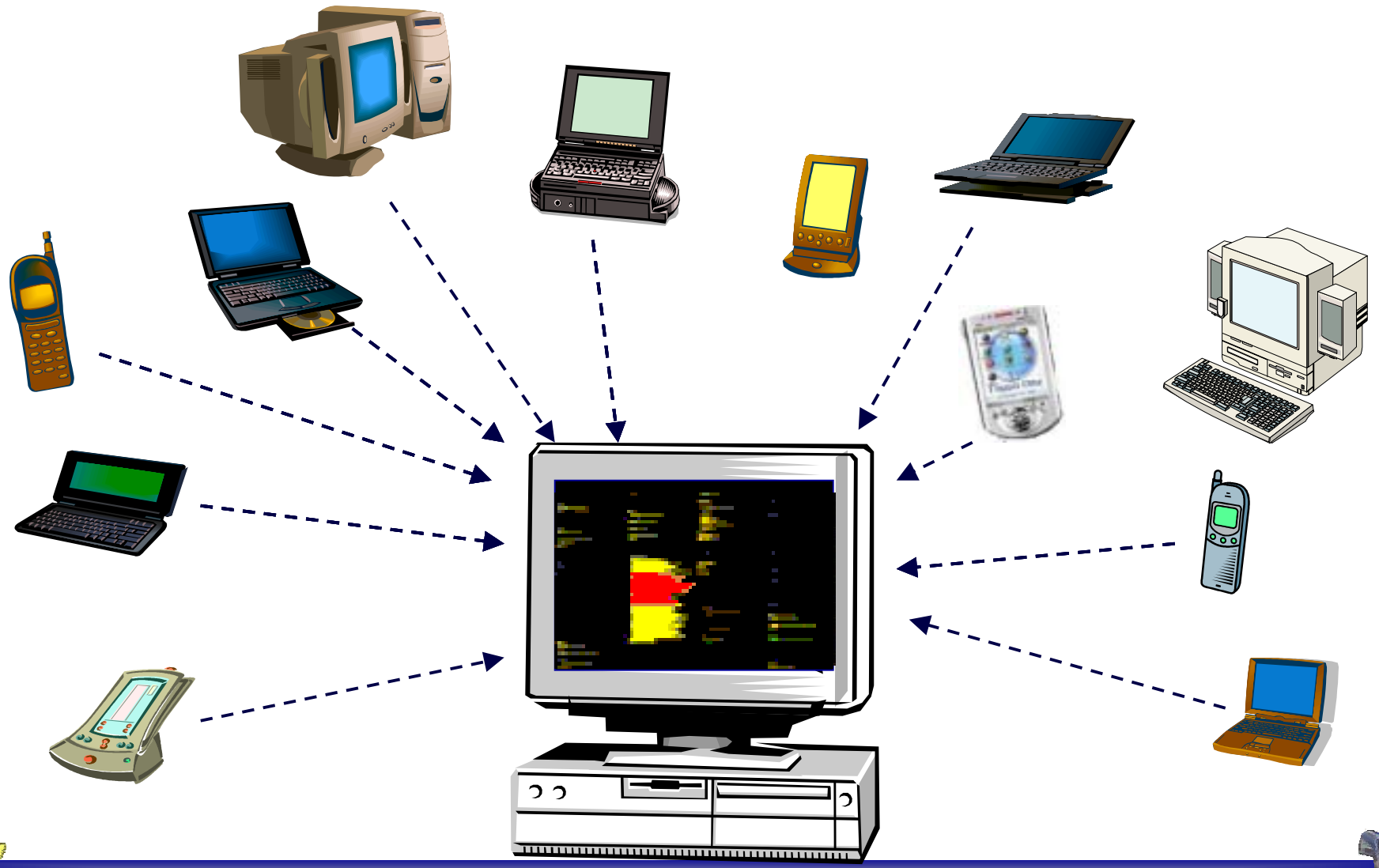
- perform continuous, minimally intrusive analysis and testing of their software in the field
- use the gathered data to respond promptly and effectively to problems and to improve and evolve their software

GAMMA: Continuous Analysis

GAMMA: Continuous Analysis



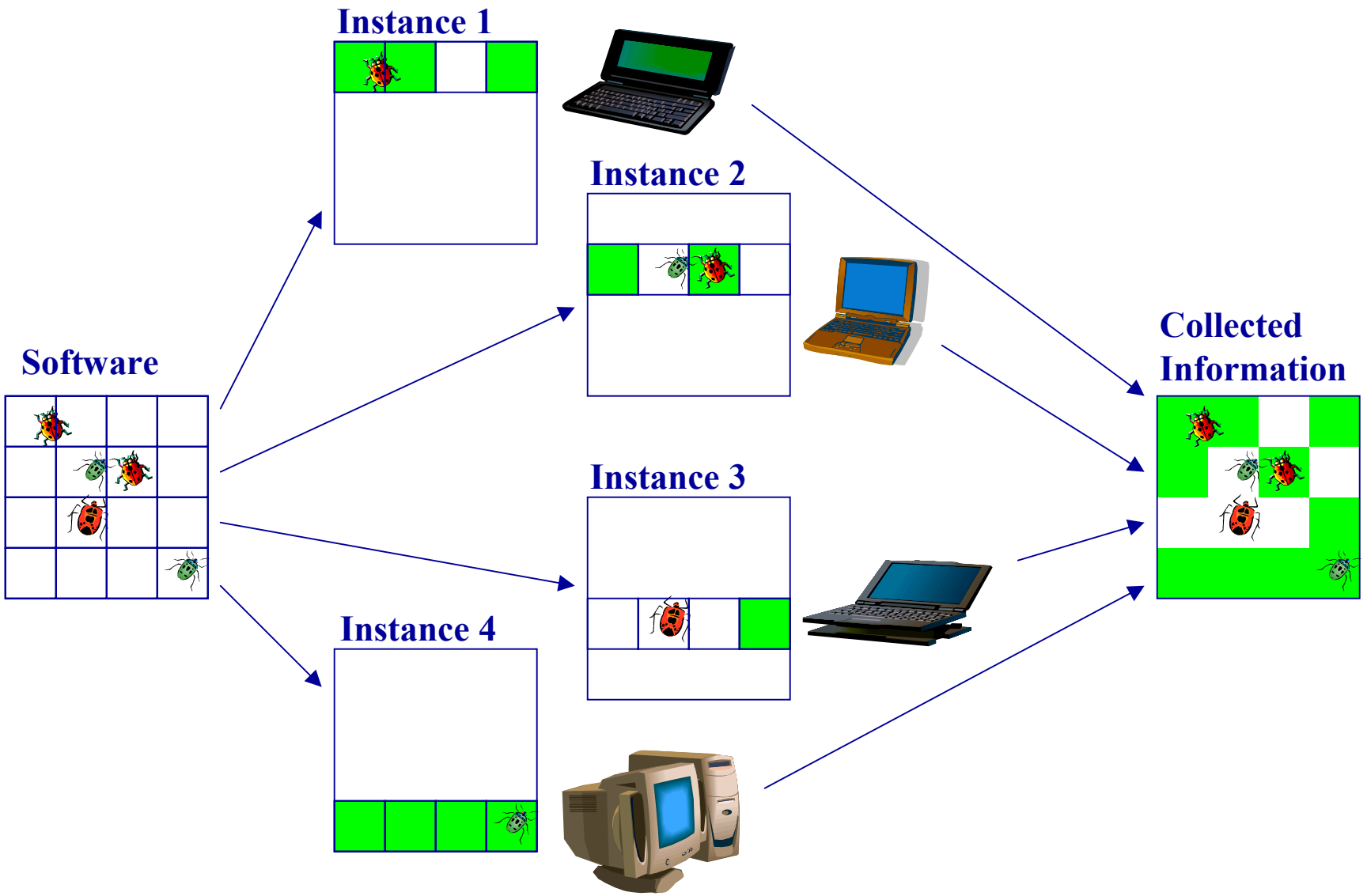
GAMMA: Continuous Analysis



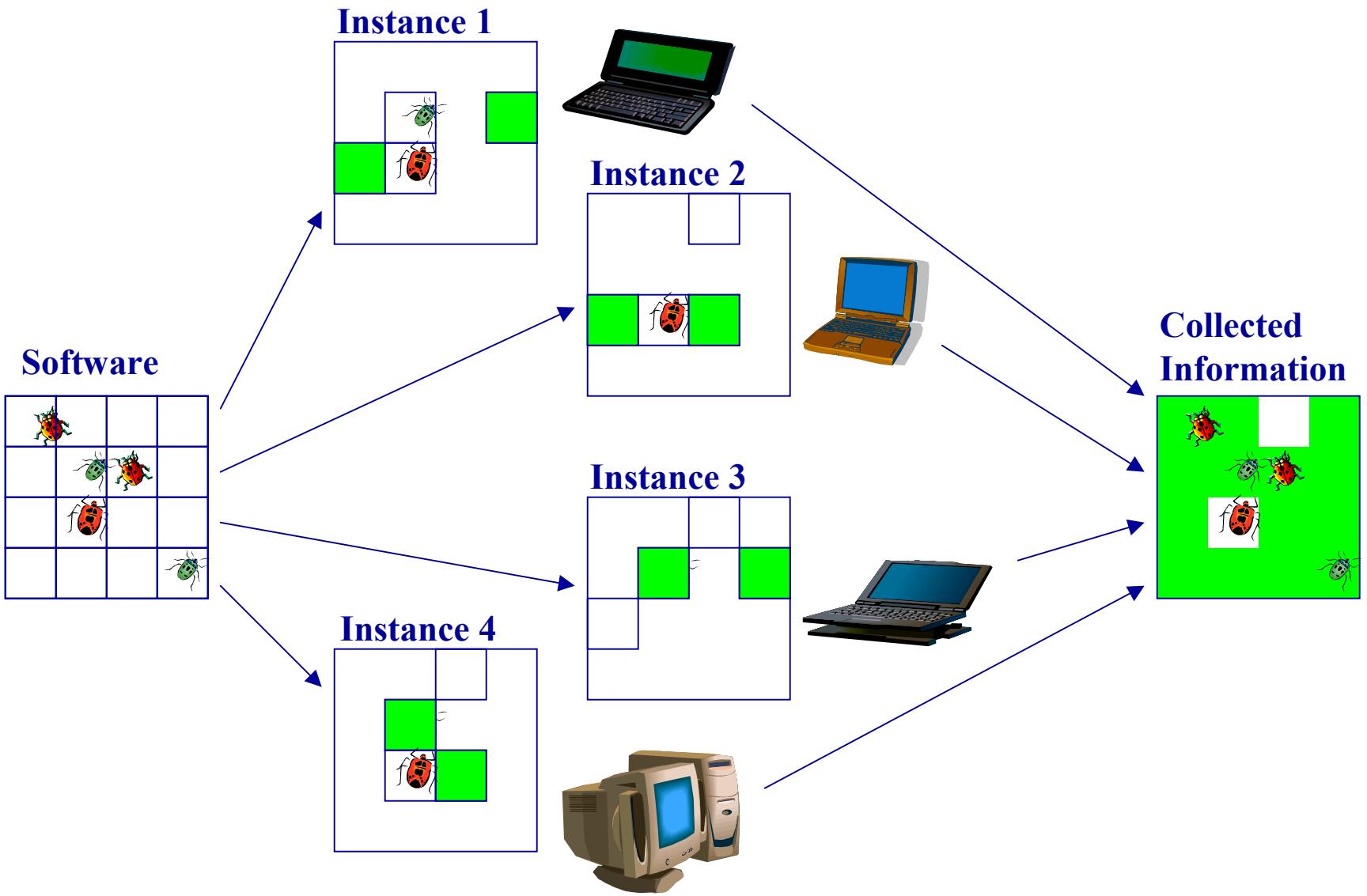
GAMMA Concepts

- Software tomography
  - Lightweight instrumentation
  - Information fusion
  - Statistical analysis
- Dynamic updating
  - Instrumentation changes
  - Software updates
- Anomaly detection
  - Failed assertions
  - Uncaught exceptions

Software Tomography



Software Tomography



Software Tomography

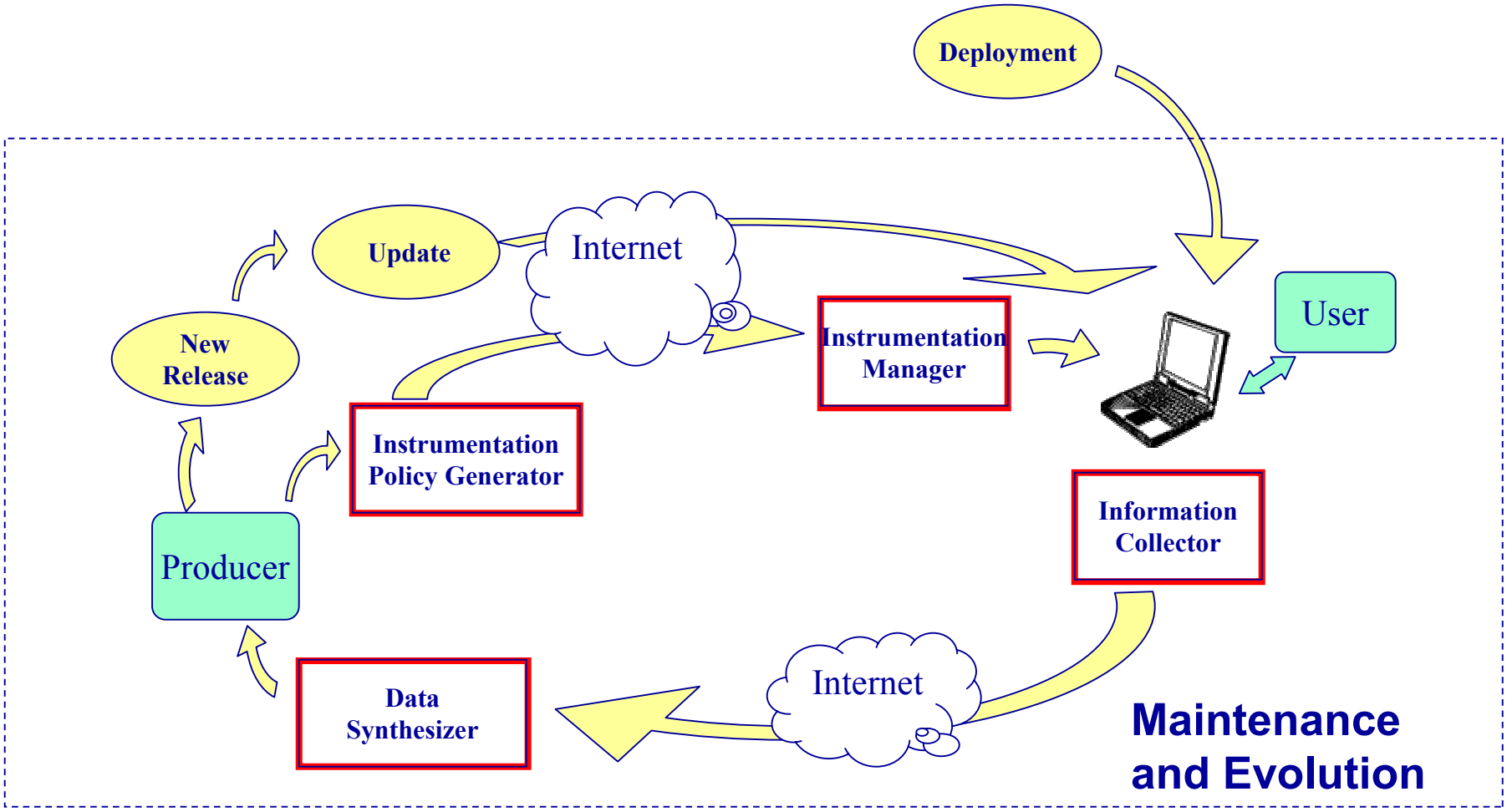
Activities

1. Subtask identification
2. Subtask assignment
3. Refinement

Tasks defined so far

- Statement coverage
- Branch coverage
- Exception coverage
- Acyclic-path coverage

GAMMA System



Experience with GAMMA

Monitoring of a desktop application

- Software tomography for structural coverage
- Redistribution for refinement
- Residual coverage
- Incremental analysis

Monitoring of hand-held devices

- Tune the technique
- Collect operational profiles

Open Issues

Scalability

- High number of users and sites possibly involved

⇒ High data volume received, stored, and analyzed

Security and Privacy

- Security in the communications between producers and users
- Privacy with respect to collected data

Related Work

Perpetual/Residual Testing (Clarke, Osterweil, Richardson, and Young)

Continuous analysis and testing of software

- The instrumentation of the residue can be too costly
- So far, statement coverage

EDEM: Expectation-driven Event Monitoring (Hilbert, Redmiles, and Taylor)

Events monitoring after deployment using agents

- Mostly limited to HCI
- Complete instrumentation

Conclusion

Contributions

- New approach for continuous monitoring of software
  - Lightweight instrumentation
  - Dynamic software update
  - Anomaly detection
- First prototype of the GAMMA system
- Initial case studies

Future work

- Extending the set of monitoring tasks considered
- Investigating off-line analyses
- Identification of abnormal software behavior
- Further evaluating the system

For more information:

http://gamma.cc.gatech.edu

Questions?