Colin J. Neill and Phillip A. Laplante
Requirements Engineering: The State of the Practice

IEEE Software **20(6)** pp. 40-45, Nov-Dec 2003

Presented by Dror Feitelson

Data

- Many people refer to "dominant", "prevalent", or "common" practices

- But there is no data to support such claims

- Set out to conduct a survey and get data

- Goal is requirements engineering, but also some more general SE data

Survey Population

- Web survey
- By invitation only
- Students in Penn State Great Valley School of Graduate Professional Studies
- School is for industrial practitioners
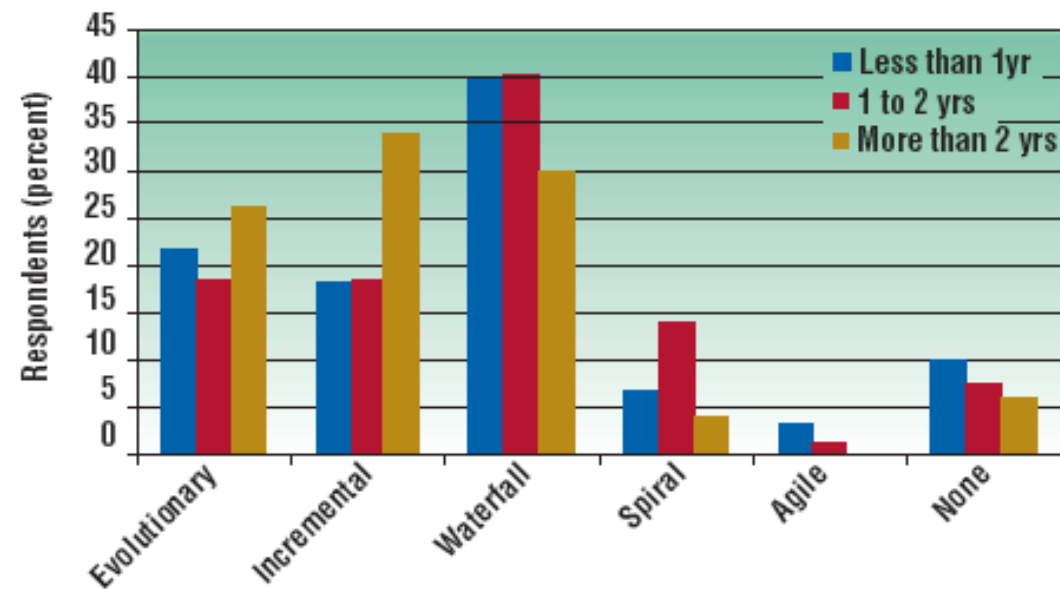- 1519 invitations
- 194 responses
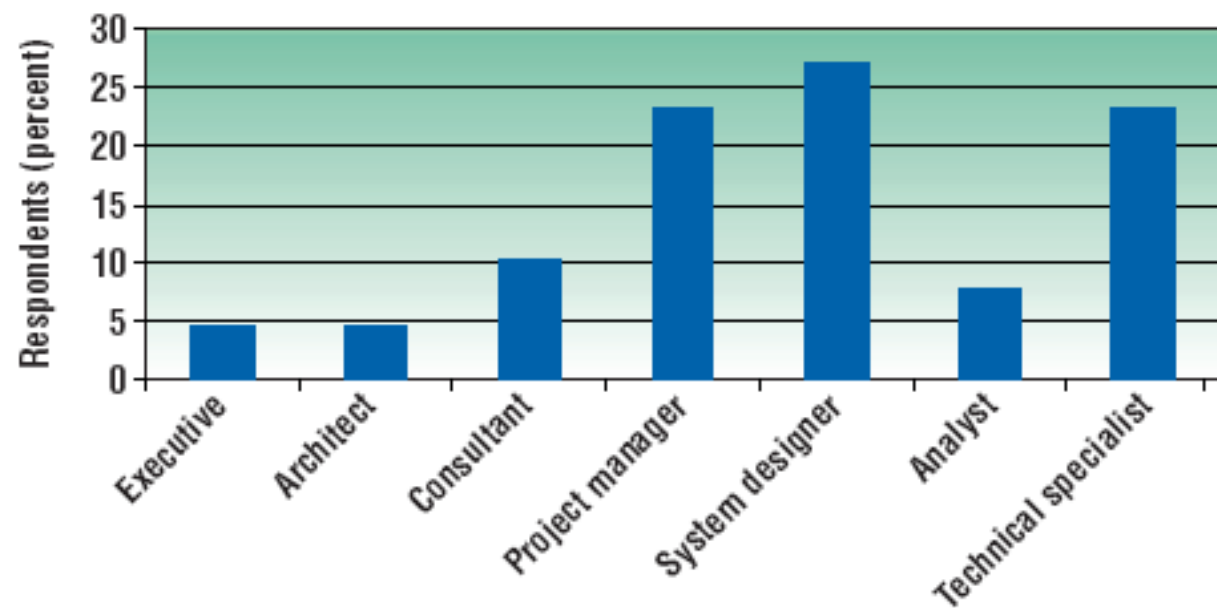
Lifecycle Model

- Maintained popularity of waterfall model

- More among developers than managers

- More in short projects than long ones

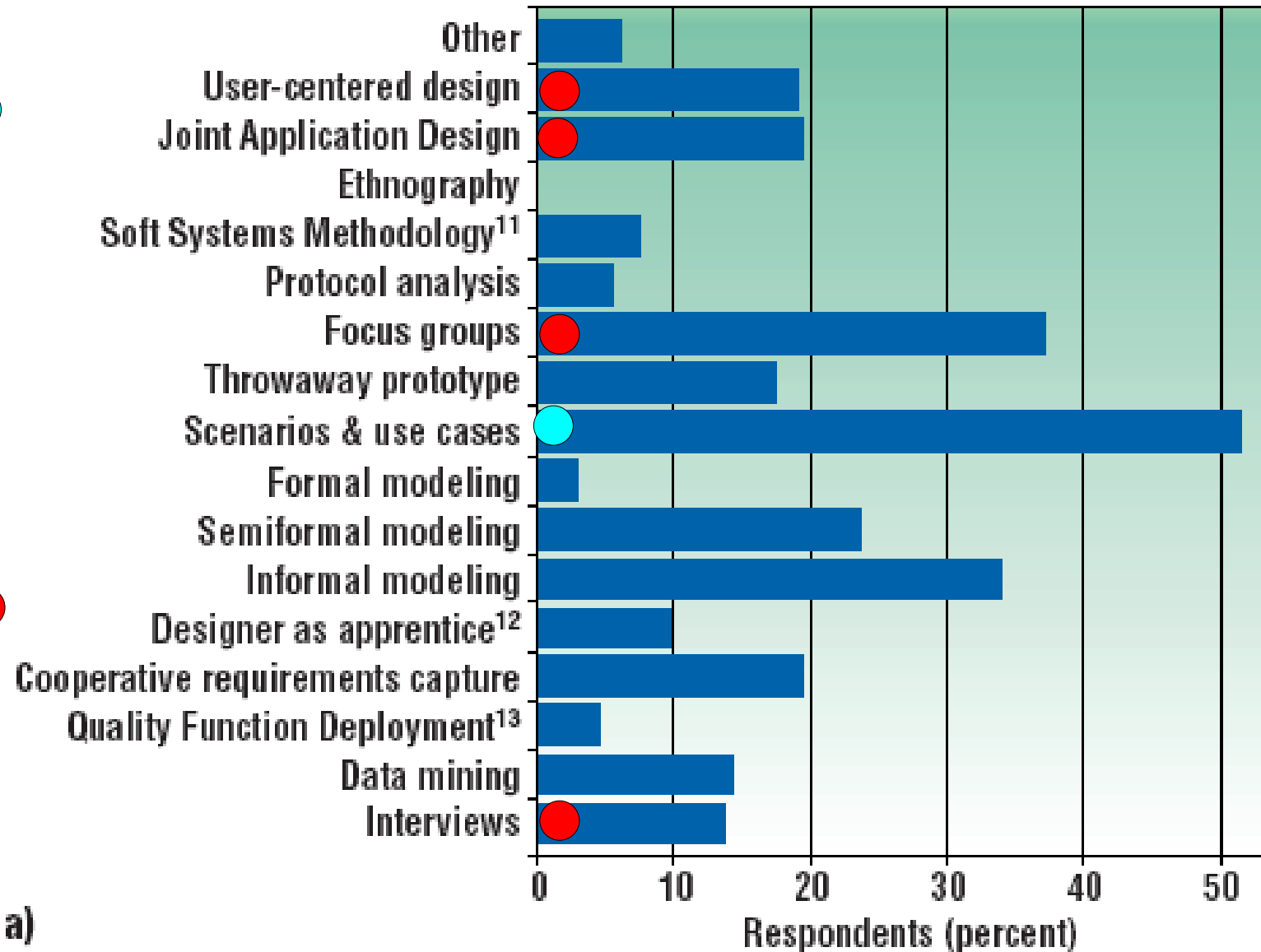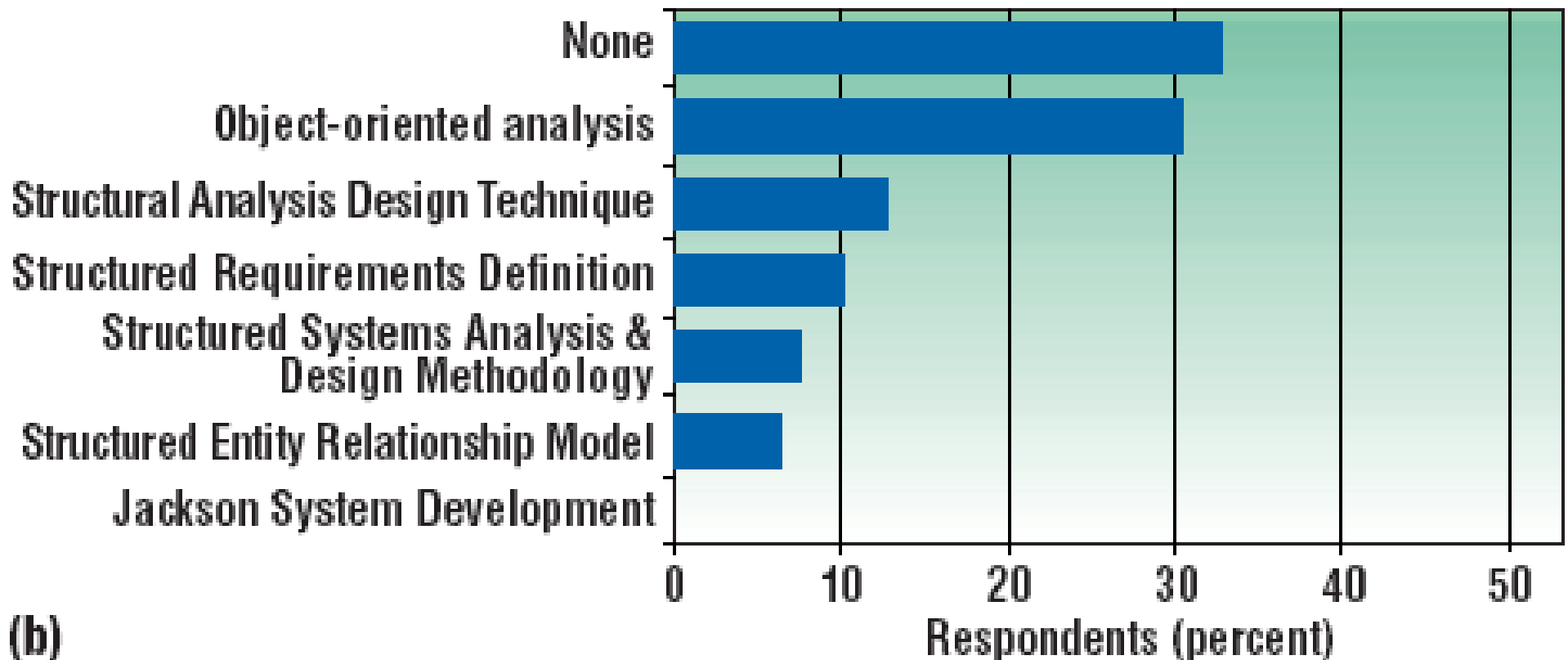- 60% used prototypes, mostly for UI, but half were evolutionary
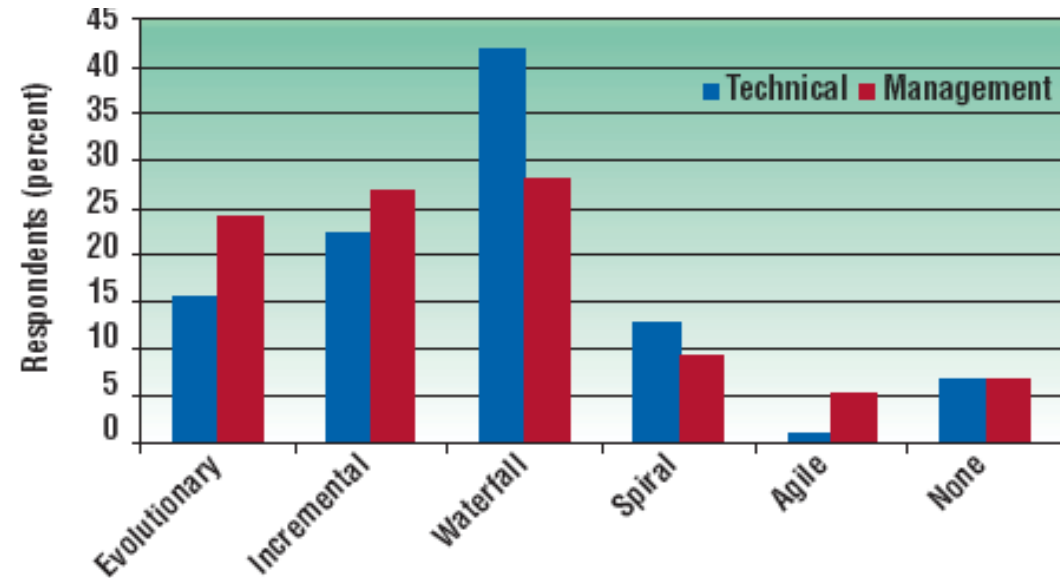
Requirements Elicitation

- Use cases (related to OO tools and methods) ○
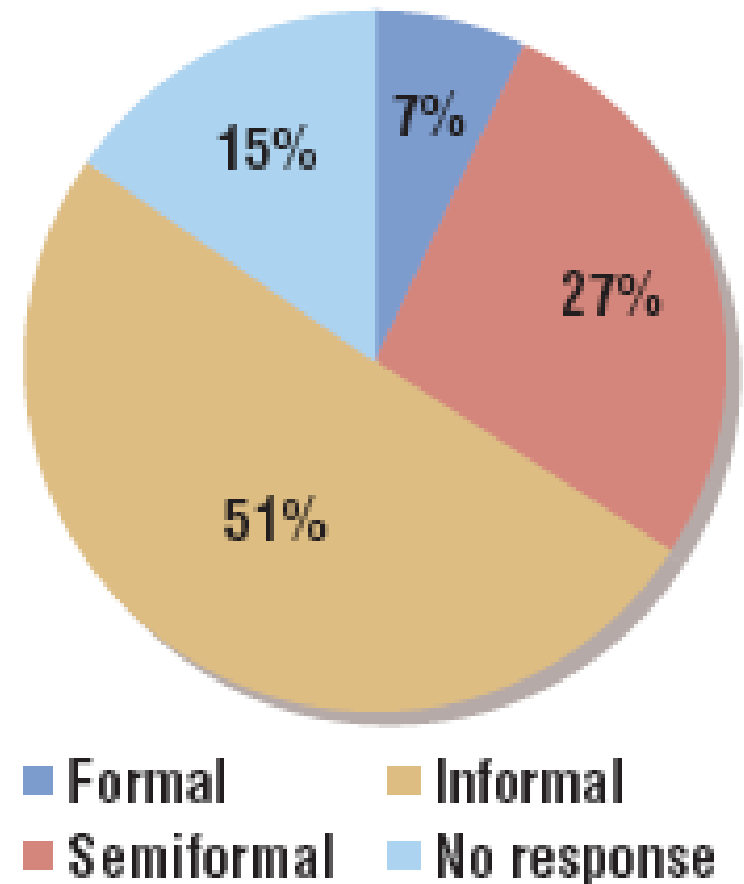
- Group consensus ●

a)

Requirements Modeling

- ~37% use specialized SE methodologies
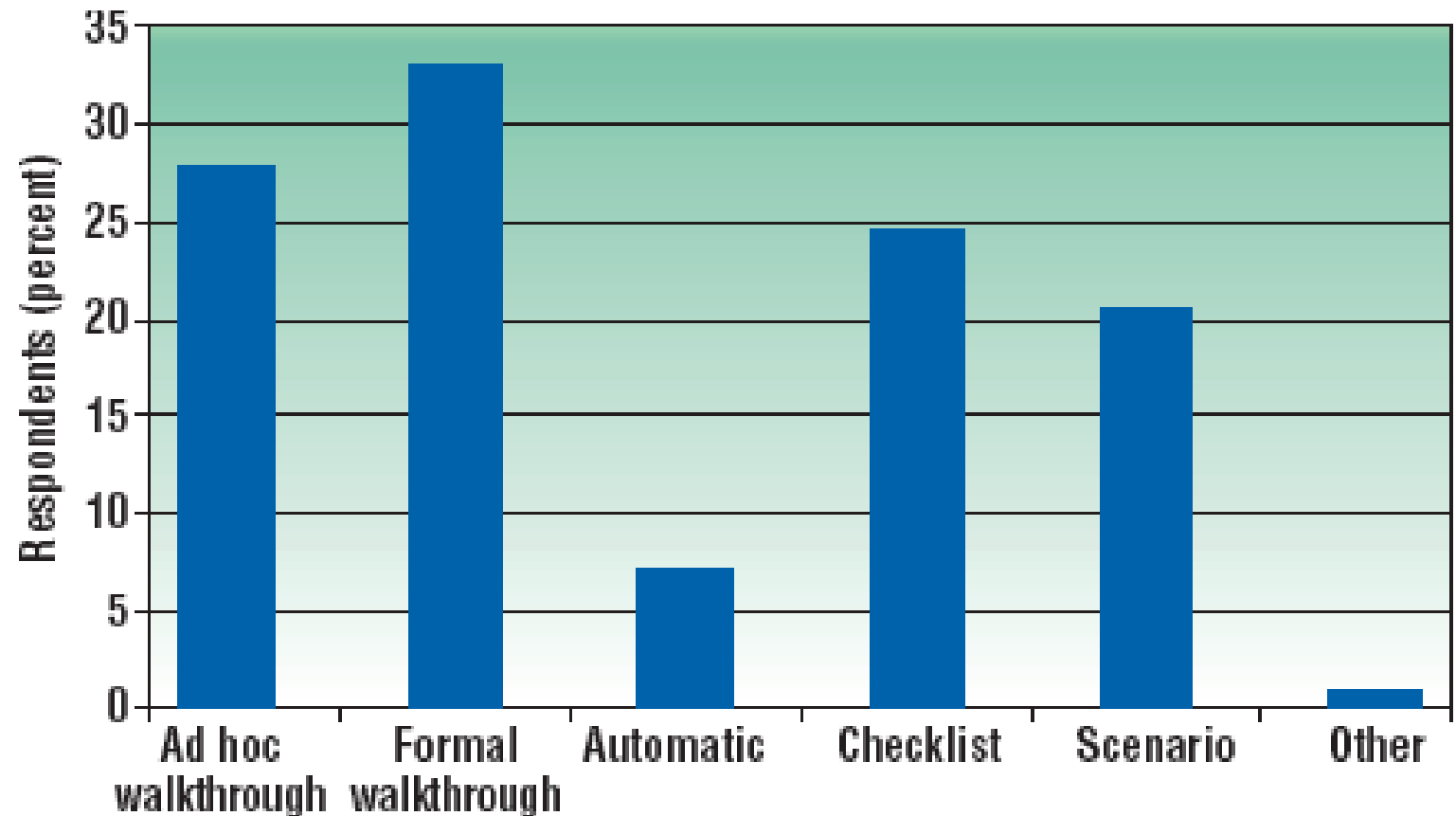- Rest use OOA or nothing

Formalism

- Requirements are usually informal (e.g. natural language)

- Higher % of informal cases report that results fit user needs and were easy to use

7%

15%

27%

51%

■ Formal   ■ Informal

■ Semiformal   ■ No response
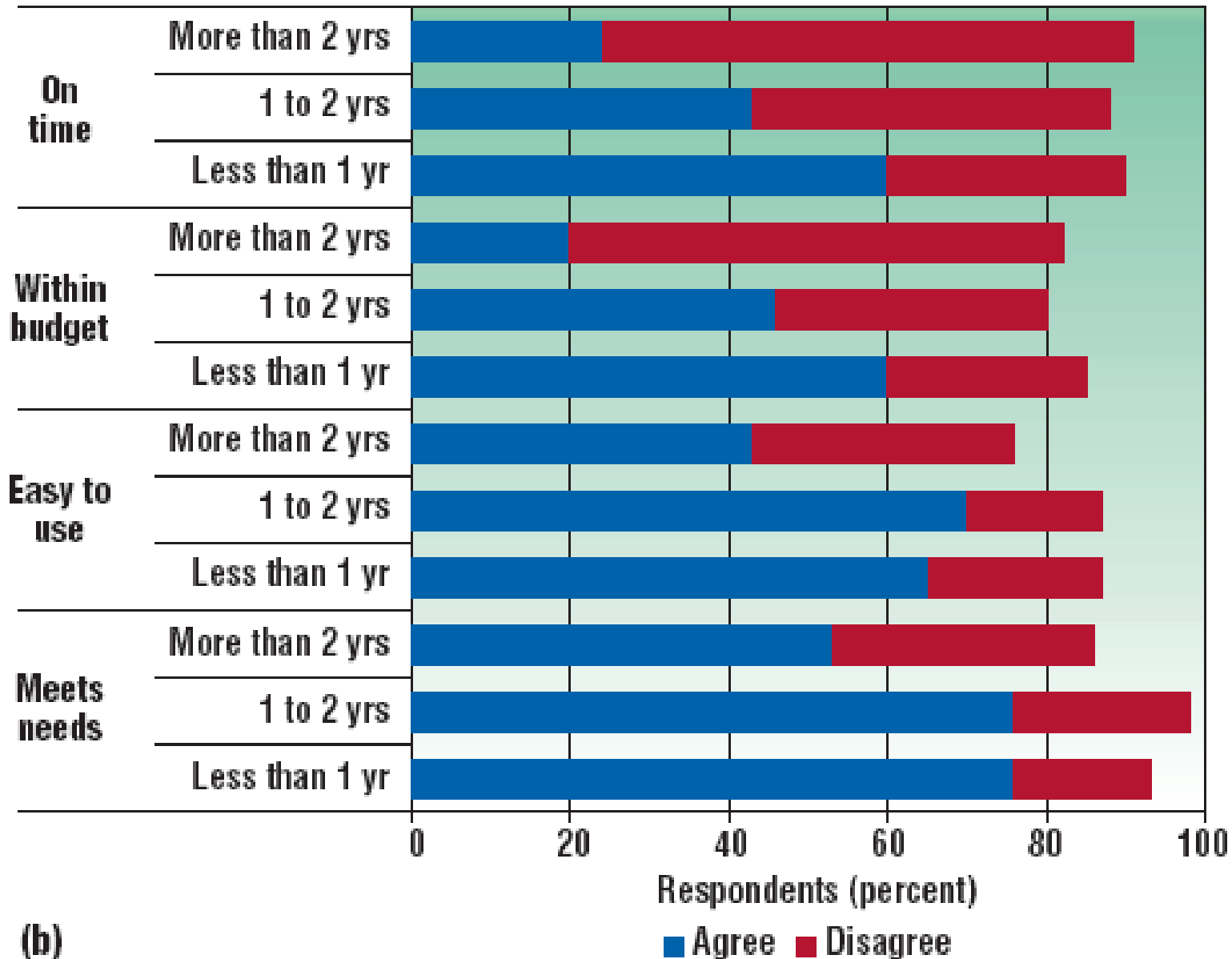
Inspection and Review

- 59% inspect the requirements
- Diverse techniques are used

Results

- Long projects are problematic

- General optimism about meeting needs

(b)

Conclusions

- Formal methods are rarely used

- Ad-hoc methods lead to good quality

- Waterfall is still popular

- OO techniques are not dominant

- Industry perception is that most projects (especially short ones) are successful

Discussion

- Is this relevant?
- Many failures attributed to wrong requirements (system works but it's the wrong system)
  - Berry: uncover inconsistent assumptions
  - Gilb: explicitly quantify qualitative requirements
- But requirements are built-in when satisfying a personal itch
- And they are discovered with progress in agile development with user participation