UNIVERSIDAD NACIONAL DEL ALTIPLANO

”FACULTAD DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA”

The Software CRISIS

Learners:
AYCAYA PACO YHACK BRYAN
MAMANI ROMERO DIEGO JHOEL
TAPARA TRISTAN DENILZON ROBINHO
TICONA CASA KATIA DAISHY

Teacher:
TORRES CRUZ FRED

Course:
SOFTWARE ENGINEERING

SEMESTER VII

PUNO, PERÚ
2024

1. Schames

- The Software Crisis
  - Description
    - The ”Software Crisis”describe a series of fundamental problems in the field of software engineering.
    - Was marked by a significant increase in software complexity and expectations of functionality and performance. during rapid computing growth.
  - Impact of the Crisis
    - Software projects often suffered significant delays, exceeded budgets, and resulted in defective or ineffective software.
    - Large-scale projects demonstrated the inadequacy of ad hoc approaches in software engineering for handling scale and complexity.
  - Responses to the Crisis
    - The software engineering community began to develop and adopt structured lifecycle models like the waterfall model.
    - Formal methods were introduced to provide a framework for specifying, developing, and verifying software within a formalized system.
  - Language
    - ALGOL
    - Pascal
    - Ada
      - Promote data abstraction and encapsulation to facilitate writing modular and maintainable code
  - Contributions
    - Served as a catalyst for the professionalization of software engineering.
    - Adoption of structured methodologies, formal methods, and professionalization.
    - The lessons learned during that period have shaped modern software engineering.
    - Its teachings remain relevant as the software industry continues to face new challenges and complexities.

2. (Keywords)

- **Software Crisis** - Period of significant software development challenges.
- **Fundamental Problems** - Core issues underlying the software crisis.
- **Software Engineering** - Systematic approach to software design and development.
- **Rapid Growth of Computing** - Swift expansion of computing technologies.
- **Software Complexity** - Intricacy of software systems.
- **Functionality and Performance Expectations** - Desired software features and efficiency levels.
- **Traditional Development Methods** - Conventional software development approaches.
- **Software Projects** - Endeavors to develop software systems.
- **High-Level Programming Languages** - Languages simplifying coding and maintenance.
- **Formal Methods** - Rigorous techniques for software specification and verification.