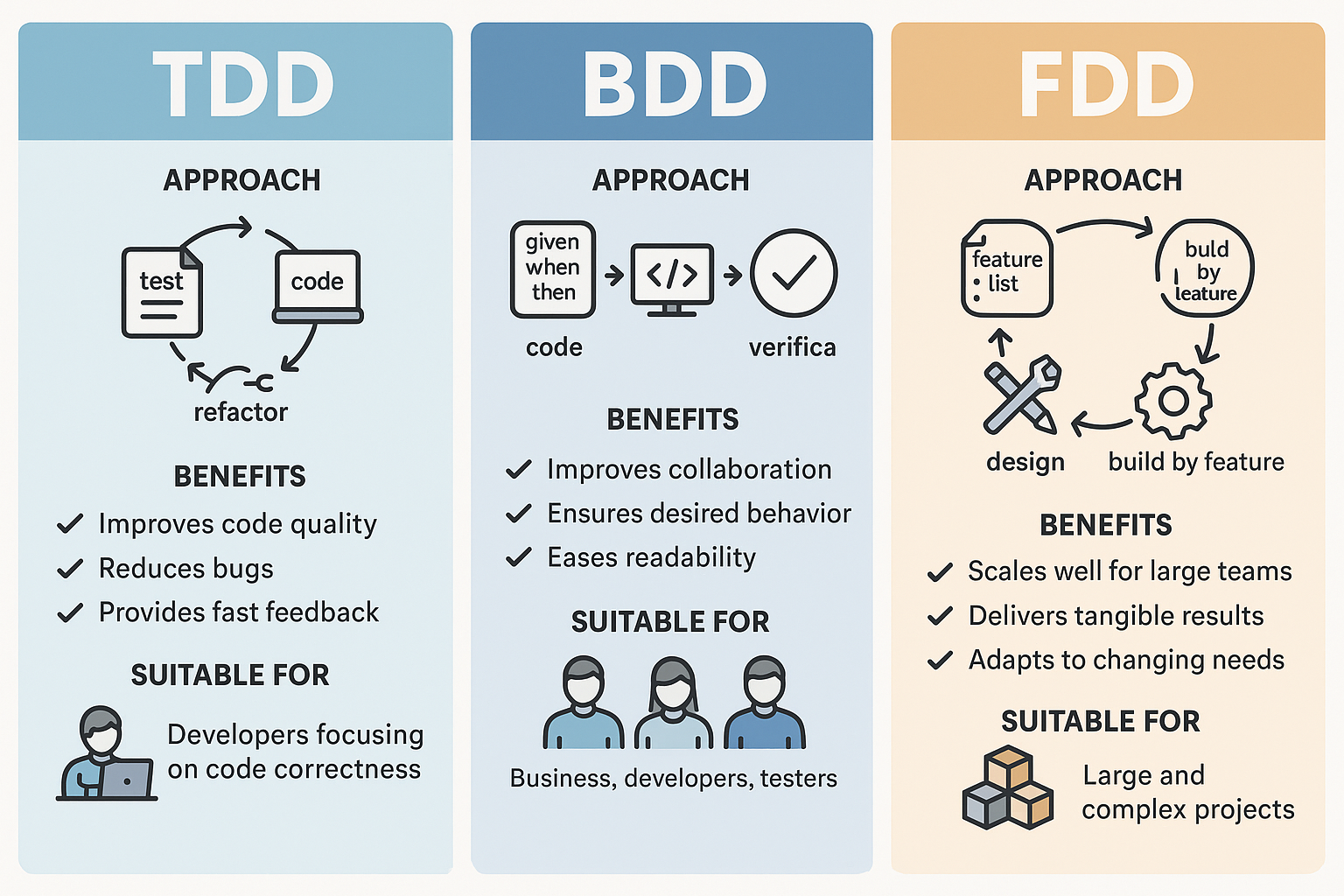
ID: 27381

ASSIGNMENT 5

Produce a comparative infographic of TDD, BDD, and FDD methodologies. Illustrate their unique approaches, benefits, and suitability for different software development contexts. Use visuals to enhance understanding.



**Overview of Each Methodology:**

**Test-Driven Development (TDD)**

* Write tests before code
* Focus on code correctness and quality
* Follows red-green-refactor cycle

**Behaviour-Driven Development (BDD)**

* Extends TDD with user behaviour focus
* Uses natural language specifications
* Emphasizes stakeholder collaboration

**Feature-Driven Development (FDD)**

* Model-driven, iterative approach
* Focuses on delivering tangible features
* Breaks down projects into manageable features

**Key Characteristics:**

**TDD:**

* Unit test-focused
* Short development cycles
* Continuous code improvement
* Immediate feedback loop

**BDD:**

* Strong stakeholder collaboration
* Natural language specifications
* User-centric approach
* Living documentation

**FDD:**

* Upfront domain modelling
* Feature-based iterations
* Clear progress tracking
* Strong project management

**Best Suited For:**

**TDD**

* Complex technical implementations
* Projects requiring high reliability
* Systems needing frequent refactoring

**BDD**

* User-centric applications
* Projects needing strong stakeholder alignment
* Systems with clear behavioural requirements

**FDD**

* Large-scale projects
* Feature-centric development
* Projects requiring clear progress tracking

**Benefits and Limitations:**

TDD Benefits

* Improved code quality
* Early bug detection
* Better design through refactoring

TDD Limitations

* Time-consuming initially
* Steep learning curve
* Can be rigid for changing requirements

BDD Benefits

* Enhanced collaboration
* Clear requirements understanding
* Better stakeholder communication

BDD Limitations

* Complex setup needed
* Requires more initial planning
* Higher performance overhead

FDD Benefits

* Scalable for large projects
* Clear progress tracking
* Strong design focus

FDD Limitations

* Less flexible than other methods
* Requires experienced teams
* Heavy on upfront planning

**Process Flow:**

**TDD Cycle**

1. Write Test → 2. Run Test (Fails) → 3. Write Code → 4. Run Test (Passes) → 5. Refactor → 6. Repeat

**BDD Process**

1. Discovery → 2. Define Scenarios → 3. Automate → 4. Develop → 5. Test → 6. Refactor → 7. Release

**FDD Lifecycle**

1. Develop Overall Model → 2. Build Features List → 3. Plan by Feature → 4. Design by Feature → 5. Build by Feature

**Key Differentiators**

* **Focus:** TDD emphasizes code quality, BDD focuses on user behaviour, FDD prioritizes feature delivery
* **Collaboration:** BDD requires the most stakeholder involvement, while TDD and FDD are more developer-centric
* **Implementation:** TDD and BDD start with tests, FDD starts with modelling and planning.